User Group & Role Administration Services

User groups and roles can be created and modified with the web service calls discussed in this section.

Note: If the Client Org functionality is switched on in the Configuration page, a Client Org can also be specified where applicable for certain types of calls.

User Role Functions

These web services are specific to Yellowfin user roles.

> ⚠ When using LDAP authentication, any web services that return a role, will return the role that the LDAP user last logged in with successfully. (This role is updated every time an LDAP user logs in.)

This function returns all the user roles available in Yellowfin. The response contains an array of AdministrationRole objects displaying available roles.

Request Elements

The following elements will be passed with this request:

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

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:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>LISTROLES</function>
         </arg0>
      </web:remoteAdministrationCall>
   </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
Roles	AdministrationRole[]	List of roles

Response Elements

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <roles>
               <functions>
                  <accessLevelCode>CRUD</accessLevelCode>
                  <functionCode>ACTIVITYSTREAM</functionCode>
                  <functionDescription>Allows users to access the activity stream.</functionDescription>
                  <functionName>Activity Stream</functionName>
               </functions>
               <functions>
                  <accessLevelCode>CRUD</accessLevelCode>
                  <functionCode>TIMELINE</functionCode>
                  <functionDescription>Allows users to access their timeline.</functionDescription>
                  <functionName>Timeline</functionName>
               </functions>
               <functions>
                  <accessLevelCode>CRUD</accessLevelCode>
                  <functionCode>BROADCASTSUBSCRIBE</functionCode>
                  <functionDescription>Allows users to subscribe to report broadcasts.</functionDescription>
                  <functionName>Subscribe to Broadcast</functionName>
               </functions>
               <functions>
                  <accessLevelCode>R</accessLevelCode>
                  <functionCode>STORYBOARD</functionCode>
                  <functionDescription>Allows users to view, create, edit or delete Storyboards.<
/functionDescription>
                  <functionName>Storyboard</functionName>
               </functions>
               <functions>
                  <accessLevelCode>R</accessLevelCode>
                  <functionCode>DASHPUBLIC</functionCode>
                  <functionDescription>Allows users to create and edit Public dashboards.</functionDescription>
                  <functionName>Public Dashboards</functionName>
               </functions>
               <functions>
                  <accessLevelCode>CRUD</accessLevelCode>
                  <functionCode>TASKPERSONAL</functionCode>
                  <functionDescription>Allow users to create and assign tasks to themselves.<
/functionDescription>
                  <functionName>Personal Tasks</functionName>
               </functions>
               .
               .
                           .
               <roleCode>YFADMIN</roleCode>
               <roleDescription>This user has the widest range of access to the system, and as such you should
have a very limited number of people assigned this role. They can do everything from create content through to
managing system tasks.</roleDescription>
               <roleName>System Administrator</roleName>
            </roles>
            <sessionId>4f86f0e30e30bf4b07dea21267de0a74</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Displayed below is a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("LISTROLES");
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE
Roles	AdministrationRole[]	List of roles

Complete Example

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

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

```
<%
/*              ws_listroles.jsp                */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin webservices admin account
rsr.setPassword("test");                           // change to be the password of the account above
rsr.setOrgId(1);

rsr.setFunction("LISTROLES");

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
        out.write("Success.<br>Available Roles:");
        AdministrationRole[] roles = rs.getRoles();
        for (AdministrationRole role: roles){
                out.write("<br>");
                out.write("<br>Role Name: " + role.getRoleName());
                out.write("<br>Role Code: " + role.getRoleCode());
                out.write("<br>Role Description: " + role.getRoleDescription());

                // uncomment to display all the security functions:
                /*
                out.write("<br>Function Name | Code | Description | TypeCode | AccessLevelCode");
                for (AdministrationFunction f: role.getFunctions()){
                        out.write("<br>"         + f.getFunctionName() + " | "
                                                 + f.getFunctionCode() + " | "
                                                 + f.getFunctionDescription() + " | "
                                                 + f.getFunctionTypeCode() + " | "
                                                 + f.getAccessLevelCode());
                }
                */

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

This function creates a new role and/or updates a role's functions. The request must contain an AdministrationRole object to specify the role details, and an array of AdministrationFunction for the role. Whether this function is used to update a role, or create a new one, it should be noted that every Yellowfin role requires a mandatory function, **Report Access** (function code: MIREPORT). MIREPORT must have its access level code set to at least **R** (read). Each time this function is called, the security functions will be overwritten.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description

LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "SAVEROLE".
Role	AdministrationRole	This object contains details of the role to be added/updated. See table below.

These are the main parameters that you need to set in the AdministrationRole object for this function:

AdministrationRole Element	Data Type	Description
RoleCode	String	To specify the internal code of an existing role. This parameter must be included if you want to update a role that already exists. If unspecified, a new role will be created, even if one with the same name already exists.
RoleName	String	Name of the new or existing role. This is mandatory even when modifying an existing role, otherwise the call will set the role name to blank.
RoleDescription	String	Description of the role.
Functions	AdministrationFunction	This object contains a list of security functions. These will be overwritten every time the Save Role function is called. The function Report Access is mandatory. See table below for more details.

These are the main parameters that you need to set in the AdministrationFunction object for this web service:

AdministrationFunction Element	Data Type	Description
FunctionCode	String	To specify the code of a security function. For example, to include the function Report Access, specify it with its code MIREPORT.
AccessLevelCode	String	The access level of the function. For example, R means read.

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:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>SAVEROLE</function>
            <role>
                    <roleCode>REPORTWRITER</roleCode>
                    <roleName>Report Content Writer</roleName>
                    <roleDescription>This role can generate reports.</roleDescription>
                    <functions>
                            <functionCode>MIREPORT</functionCode>
                            <accessLevelCode>R</accessLevelCode>
                    </functions>
            </role>
         </arg0>
      </web:remoteAdministrationCall>
   </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

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <roles>
               <functions>
                  <accessLevelCode>R</accessLevelCode>
                  <functionCode>MIREPORT</functionCode>
               </functions>
               <roleCode>REPORTCONTENTWRITER</roleCode>
               <roleDescription>This role can generate reports.</roleDescription>
               <roleName>Report Content Writer</roleName>
            </roles>
            <sessionId>ceaa85d0ca1eb6057dc4facb0a7a5aa9</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Displayed below is a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("SAVEROLE");
```

- Then define a role:

```
AdministrationRole role = new AdministrationRole();
```

- Role Code is mandatory if you want to modify an existing role:

```
role.setRoleCode("NEWROLE");                        // If you want to create a new role, comment this
out.
```

If you do not specify the Role Code, the call will create a new role even if another role with the same name already exists.

You can get the role codes from Yellowfin's database OrgRole table. (Usually, it is based on role name with all letters capitalized and with no space between them.)

- Role Name is mandatory even when modifying an existing role, otherwise the call will set the role name to blank:

```
role.setRoleName("New Role");
role.setRoleDescription("testing");
```

- Each time you call the SAVEROLE function, you need to provide a list of security functions. This call overwrites the role function. For instance, 2 functions will be assigned to the role: Report Access (mandatory) and Activity Stream (optional):

```
AdministrationFunction[] f = new AdministrationFunction[1];
f[0] = new AdministrationFunction();
f[0].setFunctionCode("MIREPORT");
f[0].setAccessLevelCode("R");
f[1] = new AdministrationFunction();
f[1].setFunctionCode("ACTIVITYSTREAM");
f[1].setAccessLevelCode("CRUD");
```


You cannot omit security functions; the call will generate an error otherwise.

- Then feed the security functions to the role:

```
role.setFunctions(f);
rsr.setRole(role);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE

Complete Example

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

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

```
<%
/*              ws_saverole.jsp               */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");                           // change to the password of the account above
rsr.setOrgId(1);

rsr.setFunction("SAVEROLE");

//define a role:
AdministrationRole role = new AdministrationRole();
role.setRoleCode("NEWROLE");
role.setRoleName("New Role");
role.setRoleDescription("testing");

AdministrationFunction[] f = new AdministrationFunction[2];

f[0] = new AdministrationFunction();
f[0].setFunctionCode("MIREPORT");           // mandatory
f[0].setAccessLevelCode("R");

f[1] = new AdministrationFunction();
f[1].setFunctionCode("ACTIVITYSTREAM");
f[1].setAccessLevelCode("CRUD");


//Feed the security functions to the role:
role.setFunctions(f);

rsr.setRole(role);
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);


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

This function deletes a specified user role. You can identify this role by providing the Role Code in the AdministrationRole object.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "DELETEROLE".
Role	AdministrationRole	This object contains details of the role to be deleted. See table below.

These are the main parameters that you need to set in the AdministrationRole object:

AdministrationRole Element	Data Type	Description
RoleCode	String	To specify the internal code of an existing role that is to be deleted.

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:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>DELETEROLE</function>
            <role>
                    <roleCode>REPORTWRITER</roleCode>
            </role>
         </arg0>
      </web:remoteAdministrationCall>
   </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

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
     <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
        <return>
           <errorCode>0</errorCode>
           <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
           <messages>Web Service Request Complete</messages>
           <roles>
              <roleCode>REPORTWRITER</roleCode>
           </roles>
           <sessionId>6c494a263bb684c1082317d0e1d695eb</sessionId>
           <statusCode>SUCCESS</statusCode>
        </return>
     </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

Instructions

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

- Displayed below is a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("DELETEROLE");
```

- Define the role that needs to be deleted:

```
AdministrationRole role = new AdministrationRole();
role.setRoleCode("NEWROLE");                    // existing role. Role Codes can be found by calling
LISTROLES
                                                                                                  // or
retrieved from the Yellowfin database table OrgRole.
rsr.setRole(role);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

  Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description

StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE

Complete Example

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

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

```
<%
/*              ws_deleterole.jsp                 */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin webservices admin account
rsr.setPassword("test");                           // change to the password of the above account above
rsr.setOrgId(1);


rsr.setFunction("DELETEROLE");


AdministrationRole role = new AdministrationRole();
role.setRoleCode("NEWROLE");
rsr.setRole(role);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);


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

User Group Functions

Web services related to user groups are defined below:

The LISTGROUPS function returns all the user groups available in Yellowfin. The response contains an array of AdministrationGroup objects representing available groups. For a list of groups belonging to a specific client, you can pass the Client Org reference ID in the call.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "LISTGROUPS".
OrgRef	String	You may include a Client Org ID to list groups belonging to a specific client. If this is not specified, then the default organization's groups will be returned.

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:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>LISTGROUPS</function>
         </arg0>
      </web:remoteAdministrationCall>
   </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
Groups	AdministrationGroup[]	List of groups

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
     <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
        <return>
           <errorCode>0</errorCode>
           <groups>
              <groupDescription>This group contains all users with the Admin role.</groupDescription>
              <groupId>11950</groupId>
              <groupMembers>
                 <internalId>5</internalId>
                 <loginId>admin@yellowfin.com.au</loginId>
              </groupMembers>
              <groupMembers>
                 <internalId>13000</internalId>
                 <loginId>binish.sheikh@yellowfin.com.au</loginId>
              </groupMembers>
              <groupName>Administrators</groupName>
           </groups>
           <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
           <messages>Web Service Request Complete</messages>
           <sessionId>79d937ead121745d93289f287d55b0ac</sessionId>
           <statusCode>SUCCESS</statusCode>
        </return>
     </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

Instructions

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

- Displayed below is a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("LISTGROUPS");
```

- Include a Client Org ID to list groups specific to that client. (If not included, then the default (that is, the Primary Org) groups will be displayed).

```
rsr.setOrgRef("org1");
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

  Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE
Groups	AdministrationGroup[]	List of groups

- You can retrieve members of each group by using AdministrationGroup.getGroupMembers(). This will retrieve an array of AdministrationGroupMember. Keep in mind that if the group has a user role as a member, it will not be retrieved. Only user accounts will be retrieved via getGroupMembers().

Complete Example

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

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

```
<%
/*              ws_listgroups.jsp               */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");                           // change to the password of the above account
rsr.setOrgId(1);

rsr.setFunction("LISTGROUPS");

//rsr.setOrgRef("org1");                               // provide org reference if required. Default org groups
will be retrieved otherwise


AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
        out.write("Success.<br>Available Groups:");
        AdministrationGroup[] groups = rs.getGroups();
        for (AdministrationGroup group: groups){
                out.write("<br>");
                out.write("<br>Group Name: " + group.getGroupName());
                out.write("<br>Group Id: " + group.getGroupId());
                out.write("<br>Group Description: " + group.getGroupDescription());
                out.write("<br>Group Status: " + group.getGroupStatus());
                out.write("<br>Group Internal Reference: " + group.getGroupInternalReference());


                // uncomment to display the members:
                /*
                out.write("<br>Members:<br>Login Id | Internal Id ");
                for (AdministrationGroupMember member: group.getGroupMembers()){
                        out.write("<br>" + member.getLoginId() + " | " + member.getInternalId() );
                }
                */
        }
} else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode());
}
%>
```

Use this function to retrieve a specified user group with its members. Group name must be provided to the request.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "GETGROUP".
Group	Administration Group	This object contains details of the user group to be retrieved. See table below.
OrgRef	String	You may include a Client Org ID to search for the group within a specific client org. If this is not specified, then the default organization's groups will be searched.

These are the main parameters that you need to set in the AdministrationGroup object for this function:

AdministrationGroup Element	Data Type	Description
GroupName	String	Specify name of the user group to retrieve its details and member list.

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:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>GETGROUP</function>
            <group>
                    <groupName>Administrators</groupName>
            </group>
         </arg0>
      </web:remoteAdministrationCall>
   </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
Group	AdministrationGroup[]	Group details with a list of its members.

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <group>
               <groupDescription>This group contains all users with the Admin role.</groupDescription>
               <groupId>11950</groupId>
               <groupMembers>
                  <internalId>5</internalId>
                  <loginId>admin@yellowfin.com.au</loginId>
               </groupMembers>
               <groupMembers>
                  <internalId>13000</internalId>
                  <loginId>binish.sheikh@yellowfin.com.au</loginId>
               </groupMembers>
               <groupName>Administrators</groupName>
               <groupStatus>OPEN</groupStatus>
            </group>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>54c5cf263f323b439c5834d1f6d8b645</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Displayed below is a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("GETGROUP");
```

- Include a Client Org ID to list groups specific to that client. (If not included, then the default (that is, the Primary Org) group will be displayed).

```
rsr.setOrgRef("org1");
```

- Provide name of a user group to retrieve its members:

```
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");
rsr.setGroup(group);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE
Group	AdministrationGroup[]	Group with members

- Use the following to retrieve members:

```
AdministrationGroupMember[] members = rs.getGroup().getGroupMembers();
```

You can use AdministrationGroupMember.getInternalId() to get the IpId of the Yellowfin account. Then pass it to a GETUSERBYIP call to retrieve the AdministrationPerson object for the user.

Complete Example

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

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

```
<%
/*              ws_getgroup.jsp                */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();


rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");                           // change to the password of the account above

rsr.setOrgId(1);

rsr.setFunction("GETGROUP");

//rsr.setOrgRef("org1");                               // provide org reference ID if required. Default org
will be searched otherwise

AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");
rsr.setGroup(group);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
        out.write("Success.<br>");
        group = rs.getGroup();

        out.write("<br>");
        out.write("<br>Group Name: " + group.getGroupName());
        out.write("<br>Group Id: " + group.getGroupId());
        out.write("<br>Group Description: " + group.getGroupDescription());
        out.write("<br>Group Status: " + group.getGroupStatus());
        out.write("<br>Group Internal Reference: " + group.getGroupInternalReference());


        // display the members:
        out.write("<br>Members:<br>Login Id | Internal Id ");
        for (AdministrationGroupMember member: group.getGroupMembers()){
                out.write("<br>" + member.getLoginId() + " | "  + member.getInternalId() );
        }
} else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode());
}
%>
```

This function creates a new user group in either a specified client org (if its reference ID is provided), or the default (primary) org. The new group details will be passed using the AdministrationGroup object. You may also provide group member detials via AdministrationGroupMember, to add them to the new group. (Note however, that these members must be existing Yellowfin users.)

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "CREATEGROUP".
Group	Administration Group	This object contains details of the user group to be added. See table below.
OrgRef	String	You may include a Client Org ID to add the new group within a specific client org. If this is not specified, then the group will be created in the default organization.

These are the main parameters that you need to set in the AdministrationGroup object for this function:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the new group.
GroupMembers	AdministrationGroupMember	This object can be used to provide details of the group members. See the table below .

These are the main parameters that you need to set in the AdministrationGroupMembers object for this function:

AdministrationGroupMembers Element	Data Type	Description
LoginId	String	The user ID of an existing Yellowfin user, to add them to this group.

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:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>CREATEGROUP</function>
            <group>
                    <groupName>Supervisors</groupName>
                    <groupMembers>
                            <loginId>admin@yellowfin.com.au</loginId>
                            <loginId>binish.sheikh@yellowfin.com.au</loginId>
                    </groupMembers>
            </group>
         </arg0>
      </web:remoteAdministrationCall>
   </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

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

```
 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>b1f1b17d503e1e11c05b72e674bc80ec</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Displayed below is a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("CREATEGROUP");
```

- To add the new group to a specific client, include its Client Org ID. (If not included, then the group will be created in the default (that is, the Primary Org) group).

```
rsr.setOrgRef("org1");
```

- Set parameters for the new group:

```
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Test Group");
```

- Inclue members to the group, for example:

```
AdministrationGroupMember[] member = new AdministrationGroupMember[2];

member[0] = new AdministrationGroupMember();
member[0].setLoginId("admin@yellowfin.com.au");

member[1] = new AdministrationGroupMember();
member[1].setLoginId("john.smith@yellowfin.com.au");

group.setGroupMembers(member);

rsr.setGroup(group);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE

Complete Example

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

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

```
<%
/*              ws_creategroup.jsp                */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("CREATEGROUP");

//Specify client org (if omitted, the group will be created in the default (primary) org):
rsr.setOrgRef("org1");

//Set parameters of the new group:
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Test Group");                          // mandatory. Other parameters are optional.

//Add members:
AdministrationGroupMember[] member = new AdministrationGroupMember[2];
member[0] = new AdministrationGroupMember();
member[0].setLoginId("admin@yellowfin.com.au");

member[1] = new AdministrationGroupMember();
member[1].setLoginId("john.smith@yellowfin.com.au");

group.setGroupMembers(member);

rsr.setGroup(group);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

This function is used to update the members of a group. If a list of members is provided with this request, the previous member list will be overwritten, that is, the service will delete all existing members and add the new ones. If a member list is not supplied, then all the existing members will be removed from the group.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description

LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "MODIFYGROUP".
Group	AdministrationGroup	This object contains details of the user group to be modified. See table below.
OrgRef	String	You may include a Client Org ID to search for the group within a specific client org. If this is not specified, then the group will be searched in the default (Primary) organization.

These are the main parameters that you need to set in the AdministrationGroup object for this function:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the group.
GroupMembers	AdministrationGroupMember	This object can be used to provide details of the group members. See the table below .

These are the main parameters that you need to set in the AdministrationGroupMembers object for this function:

AdministrationGroupMembers Element	Data Type	Description
LoginId	String	The user ID of an existing Yellowfin user, to add them to this group.

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:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>MODIFYGROUP</function>
            <group>
                    <groupName>Supervisors</groupName>
                    <groupMembers>
                            <loginId>admin@yellowfin.com.au</loginId>
                    </groupMembers>
            </group>
         </arg0>
      </web:remoteAdministrationCall>
   </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

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

```
 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>6589bf668504fd3468e0b43844550a22</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Displayed below is a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("MODIFYGROUP");
```

- To search for a group in a client organization, provide its Client Org ID. (If not included, then the group will be searched in the default (primary) org.)

```
rsr.setOrgRef("org1");
```

- Set parameters for the new group:

```
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Test Group");
```

- Include members to the group:

```
AdministrationGroupMember[] member = new AdministrationGroupMember[1];

member[0] = new AdministrationGroupMember();
member[0].setLoginId("admin@yellowfin.com.au");

group.setGroupMembers(member);

rsr.setGroup(group);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE

Complete Example

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

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

```
<%
/*              ws_modifygroup.jsp                */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();

AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("MODIFYGROUP");

//Specify client org (if omitted, default (primary) org will be searched):
rsr.setOrgRef("org1");

//Set parameters of the new group:
AdministrationGroup group = new AdministrationGroup();

group.setGroupName("Test Group");              // mandatory. Other parameters are optional.


//Add members:
AdministrationGroupMember[] member = new AdministrationGroupMember[1];

member[0] = new AdministrationGroupMember();
member[0].setLoginId("admin@yellowfin.com.au");

group.setGroupMembers(member);

rsr.setGroup(group);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

This function is used to rename a group. Use the AdministrationGroup object to specify the group with its ID. The group IDs can be retrieved from Yellowfin's database (**AccessGroupId** field of **AccessGroup** table) or calling GETGROUP by group name and getting response.getGroup().getGroupId().

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description

LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "RENAMEGROUP".
Group	Administration Group	This object contains details of the user group to be renamed. See table below.
OrgRef	String	You may include a Client Org ID to add the new group within a specific client org. If this is not specified, then the group will be created in the default organization.

These are the main parameters that you need to set in the AdministrationGroup object for this function:

AdministrationGroup Element	Data Type	Description
GroupId	Integer	Internal ID of the group to identify it.
GroupName	String	New name of the group.
GroupDescription	String	Description of the group.

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:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>RENAMEGROUP</function>
            <group>
               <groupId>13001</groupId>
                    <groupName>Report Creators</groupName>
                    <groupDescription>Users of this group will create reports.</groupDescription>
            </group>
         </arg0>
      </web:remoteAdministrationCall>
   </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

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

```
 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>2ca79b1696913aa7a4f8b601ac1641a4</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Displayed below is a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("RENAMEGROUP");
```

- To search for a group within a specific client, include its Client Org ID. (If not included, then the group will be searched in the default (that is, the Primary Org) group).

```
rsr.setOrgRef("org1");
```

- Define a group to rename it:

```
AdministrationGroup group = new AdministrationGroup();
```

- Identify the group by its group ID:

```
group.setGroupId(13002);
```

- Provide a new group name and description:

```
group.setGroupName("Org 1");
group.setGroupDescription("Organization 1 user group");

rsr.setGroup(group);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE

Complete Example

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

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

```
<%
/*              ws_renamegroup.jsp                */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number

AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");                           //password of the account above
rsr.setOrgId(1);

rsr.setFunction("RENAMEGROUP");

rsr.setOrgRef("org1");                       // provide org reference if required. Default org will be
searched otherwise

AdministrationGroup group = new AdministrationGroup();

group.setGroupId(13002);                          // identify the group to rename
group.setGroupName("Org1");                       // new group name

group.setGroupDescription("Organization 1 user group");              // new description

rsr.setGroup(group);


AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

Call this web service to delete an existing user group from Yellowfin, by providing the group name.

Request Parameters

The following parameters will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.

OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "DELETEGROUP".
Group	AdministrationGroup	This object contains details of the user group to be deleted. See table below.
OrgRef	String	You may include a Client Org ID to search for this group within a specific client org. If this is not specified, then the group will be searched in the default organization.

These are the main parameters that you need to set in the AdministrationGroup object for this function:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the group to be deleted.

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:remoteAdministrationCall>
         <arg0>
           <loginId>admin@yellowfin.com.au</loginId>
           <password>test</password>
           <orgId>1</orgId>
           <function>DELETEGROUP</function>
           <group>
                   <groupName>Admin</groupName>
           </group>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

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

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
     <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
        <return>
           <errorCode>0</errorCode>
           <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
           <messages>Web Service Request Complete</messages>
           <sessionId>db18f2503e80ca02a9d37da13fc540a5</sessionId>
           <statusCode>SUCCESS</statusCode>
        </return>
     </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

Instructions

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

- Start with a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("DELETEGROUP");
```

- You may specify a client organization to search for the group within it. But if not included, then the group will be searched in the default (that is, the primary) organization.

```
rsr.setOrgRef("org1");
```

- Set parameters for the group which is to be deleted:

```
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Test Group");

rsr.setGroup(group);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

  Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE

Complete Example

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

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

```
<%
/*              ws_deletegroup.jsp               */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number

AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");                           // set the password of the above account

rsr.setOrgId(1);

rsr.setFunction("DELETEGROUP");

rsr.setOrgRef("org1");                                // specify a client org reference if required. Or skip
this to search through the default org


AdministrationGroup group = new AdministrationGroup();

group.setGroupName("Test Group");                        // this group must exist in the specified client org


rsr.setGroup(group);


AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

This function is used to add a specific Yellowfin user to a specific user group.

This request will require the AdministrationPerson object to specify the user, and the AdministrationGroup object to define the user group.

Request Parameters

The following parameters will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "INCLUDEUSERINGROUP".
Person	Administration Person	Object containing details of the user that is to be added to the group. See table below.
Group	Administration Group	Object containing details of the user group to which the user is added. See table below.
OrgRef	String	You may include a Client Org ID to search for this group within a specific client org. If this is not specified, then the group will be searched in the default organization.

These are the main parameters that you need to set in the AdministrationPerson object for this function:

AdministrationPerson Element	Data Type	Description
UserId	String	An existing Yellowfin user to add them to the group. This could be a user ID or an email address, depending on the Logon method.

These are the main parameters that you need to set in the AdministrationGroup object for this function:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the group to which the user is to be added.

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:remoteAdministrationCall>
         <arg0>
           <loginId>admin@yellowfin.com.au</loginId>
           <password>test</password>
           <orgId>1</orgId>
           <function>INCLUDEUSERINGROUP</function>
           <person>
                   <userId>binish.sheikh@yellowfin.com.au</userId>
           </person>
           <group>
                   <groupName>Administrators</groupName>
           </group>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

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

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>a26d9c279a9c1a4f0dfda86424ca4267</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Start with a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("INCLUDEUSERINGROUP");
```

- You may specify a client organization to search for the group within it. But if not included, then the group will be searched in the default (that is, the primary) organization.

```
rsr.setOrgRef("org1");
```

- Set parameters to identify an existing user to include them to a group:

```
AdministrationPerson ap = new AdministrationPerson();
ap.setUserId("john.smith@yellowfin.com.au");  // must be an existing Yellowfin user

rsr.setPerson(ap);
```

- Set parameters for the group to which to include the user:

```
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");

rsr.setGroup(group);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE

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_includeuseringroup.jsp.
2. Put the file in the root folder: *Yellowfin/appserver/webapps/ROOT*.

3. Adjust the host, port, admin user, existing user, and group name according to your environment.
4. Run *http://<host>:<port>/ws_includeuseringroup.jsp* from your Internet browser.

```
<%
/*             ws_includeuseringroup.jsp             */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number

AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();

AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");                           // set to password of the above account

rsr.setOrgId(1);

rsr.setFunction("INCLUDEUSERINGROUP");


//Specify a client org (if omitted, default (primary) org groups will be searched):
rsr.setOrgRef("org1");


//Identify a user:
AdministrationPerson ap = new AdministrationPerson();
ap.setUserId("john.smith@yellowfin.com.au");                          // must be an existing Yellowfin user

rsr.setPerson(ap);



//Specify group to add the user to
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");                         // must be an existing user group

rsr.setGroup(group);


AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);


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

This function is used to include multiple specified Yellowfin users to a specific user group.

This request will require an array of AdministrationPerson object to specify the users, and the AdministrationGroup object to define the user group.

Request Parameters

The following parameters will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "INCLUDEUSERSINGROUP".
Person	Administration Person[]	An object array containing details of the users to add them to a group. See table below.
Group	Administration Group	Object containing details of the user group to which the users are added. See table below.
OrgRef	String	You may include a Client Org ID to search for this group within a specific client org. If this is not specified, then the group will be searched in the default organization.

These are the main parameters that you need to set in the AdministrationPerson object array for this function:

AdministrationPerson Element	Data Type	Description
UserId	String	Existing Yellowfin user to add them to the group. This could be a user ID or an email address, depending on the Logon ID method.

These are the main parameters that you need to set in the AdministrationGroup object for this function:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the group to which the users are to be added.

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:remoteAdministrationCall>
         <arg0>
           <loginId>admin@yellowfin.com.au</loginId>
           <password>test</password>
           <orgId>1</orgId>
           <function>INCLUDEUSERSINGROUP</function>
           <people>
                   <userId>binish.sheikh@yellowfin.com.au</userId>
                   <userId>admin@yellowfin.com.au</userId>
           </people>
           <group>
                   <groupName>Administrators</groupName>
           </group>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

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

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>799b3c35c5359c6105586e426f1b9f8c</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Start with a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("INCLUDEUSERSINGROUP");
```

- You may specify a client organization to search for the group within it. But if not included, then the group will be searched in the default (that is, the primary) organization.

```
rsr.setOrgRef("org1");
```

- Set parameters to identify existing users to include them to a group:

```
AdministrationPerson[] ap = new AdministrationPerson[1];
ap[0] = new AdministrationPerson();
ap[0].setUserId("john.smith@yellowfin.com.au");             // must be an existing Yellowfin user


rsr.setPeople(ap);
```

- Set parameters for the group where the users are to be included:

```
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");                                    //must be an existing group

rsr.setGroup(group);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE

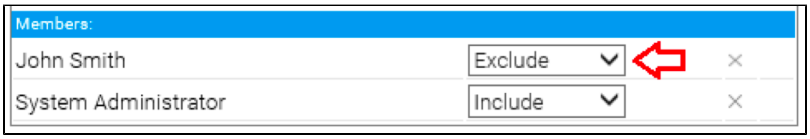
Complete Example

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_includeusersingroup.jsp.
2. Put the file in the root folder: *Yellowfin/appserver/webapps/ROOT*.
3. Adjust the host, port, admin user, existing users, and group name according to your environment.
4. Run *http://<host>:<port>/ws_includeusersingroup.jsp* from your Internet browser.

```
<%
/*             ws_includeusersingroup.jsp             */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number

AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();

AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");                           // set to password of the above account

rsr.setOrgId(1);

rsr.setFunction("INCLUDEUSERINGROUP");


//Specify a client org (if omitted, default (primary) org groups will be searched):
rsr.setOrgRef("org1");


//Provide all the users that are to be included:
AdministrationPerson[] ap = new AdministrationPerson[1];
ap[0] = new AdministrationPerson();
ap[0].setUserId("john.smith@yellowfin.com.au");                          // must be an existing Yellowfin user

rsr.setPerson(ap);



//Specify group to add the users to
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");                          // must be an existing user group

rsr.setGroup(group);


AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);


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

This function is used to add a specific Yellowfin user to a specific user group, but with an "exclude" tag. Note that this user is not actually removed from the group, but will exist as an excluded member.



This request will require the AdministrationPerson object to specify the user, and the AdministrationGroup object to define the user group.

Request Parameters

The following parameters will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "EXCLUDEUSERFROMGROUP".
Person	Administration Person	Object containing details of the user to be excluded from group. See table below.
Group	Administration Group	Object containing details of the user group. See table below.
OrgRef	String	You may include a Client Org ID to search for this group within a specific client org. If this is not specified, then the group will be searched in the default organization.

These are the main parameters that you need to set in the AdministrationPerson object for this function:

AdministrationPerson Element	Data Type	Description
UserId	String	An existing Yellowfin user to exclude them from the group. This value could be a user ID or an email address, depending on the Logon ID method.

These are the main parameters that you need to set in the AdministrationGroup object for this function:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the group from which the user is to be excluded.

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:remoteAdministrationCall>
         <arg0>
           <loginId>admin@yellowfin.com.au</loginId>
           <password>test</password>
           <orgId>1</orgId>
           <function>EXCLUDEUSERFROMGROUP</function>
           <person>
                   <userId>binish.sheikh@yellowfin.com.au</userId>
           </person>
           <group>
                   <groupName>Administrators</groupName>
           </group>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

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

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>c15a0993df4f37f4dbff9b3244f41ea2</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Start with a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("EXCLUDEUSERFROMGROUP");
```

- You may specify a client organization to search for the group within it. But if not included, then the group will be searched in the default (that is, the primary) organization.

```
rsr.setOrgRef("org1");
```

- Set parameters to identify a user :

```
AdministrationPerson ap = new AdministrationPerson();
ap.setUserId("john.smith@yellowfin.com.au");          // must be an existing Yellowfin user

rsr.setPerson(ap);
```

- Set parameters for the group:

```
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");                       //must be an existing user group

rsr.setGroup(group);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE

Complete Example

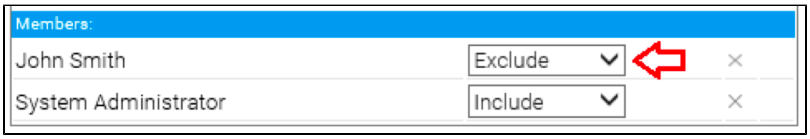
Below is a full example of this function. To use it for yourself, carry out the following the steps:

1. Copy the code and save it as ws_excludeuserfromgroup.jsp.
2. Put the file in the root folder: *Yellowfin/appserver/webapps/ROOT*.

3. Adjust the host, port, admin user, user to exclude, and the group name according to your environment.
4. Run *http://<host>:<port>/ws_excludeuserfromgroup.jsp* from your Internet browser.

```
<%
/*              ws_ excludeuserfromgroup.jsp              */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number

AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();

AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");                           // set the password of the above account

rsr.setOrgId(1);

rsr.setFunction("EXCLUDEUSERFROMGROUP");


//Specify the client org (if omitted, the default (primary) org groups will be searched):


rsr.setOrgRef("org1");


//Specify a user to exclude:


AdministrationPerson ap = new AdministrationPerson();
ap.setUserId("john.smith@yellowfin.com.au");                 // must be an existing Yellowfin use

rsr.setPerson(ap);


//Specify which group to exclude from:

AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");                // must be an existing user group

rsr.setGroup(group);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);


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

This function is used to add a specific multiple users to a specific user group, but with an "exclude" tag. Note that these users are not actually removed from the group, but will exist as an excluded members.



This request will require the AdministrationPerson object array to specify the users, and the AdministrationGroup object to define the user group.

Request Parameters

The following parameters will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "EXCLUDEUSERSFROMGROUP".
Person	Administration Person[]	Object array containing details of the users who are to be excluded from the group. See table below.
Group	Administration Group	Object containing details of the user group. See table below.
OrgRef	String	You may include a Client Org ID to search for this group within a specific client org. If this is not specified, then the group will be searched in the default organization.

These are the main parameters that you need to set in the AdministrationPerson object array for this function:

AdministrationPerson Element	Data Type	Description
UserId	String	Existing Yellowfin user to exclude them from the group. This could be a user ID or an email address, depending on the Logon method.

These are the main parameters that you need to set in the AdministrationGroup object for this function:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the group from which the users are excluded.

Response Parameters

The response returned will contain these parameters:

Response Element	Data Type	Description

StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE

Instructions

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

- Start with a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("EXCLUDEUSERSFROMGROUP");
```

- You may specify a client organization to search for the group within it. But if not included, then the group will be searched in the default (that is, the primary) organization.

```
rsr.setOrgRef("org1");
```

- Set parameters to identify a users:

```
AdministrationPerson[] ap = new AdministrationPerson[1];
ap[0] = new AdministrationPerson();
ap[0].setUserId("john.smith@yellowfin.com.au");          // must be an existing Yellowfin user

rsr.setPerson(ap);
```

- Set parameters for the group:

```
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");                        //must be an existing user group

rsr.setGroup(group);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE

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_excludeusersfromgroup.jsp.
2. Put the file in the root folder: *Yellowfin/appserver/webapps/ROOT*.
3. Adjust the host, port, admin user, users to exclude, and the group name according to your environment.
4. Run *http://<host>:<port>/ws_excludeusersfromgroup.jsp* from your Internet browser.

```
<%
/*              ws_ excludeusersfromgroup.jsp               */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number

AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();

AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");                           // set the password of the above account

rsr.setOrgId(1);

rsr.setFunction("EXCLUDEUSERFROMGROUP");


//Specify the client org (if omitted, the default (primary) org groups will be searched):


rsr.setOrgRef("org1");


//Specify users to exclude:

AdministrationPerson[] ap = new AdministrationPerson[1];
ap[0] = new AdministrationPerson();
ap[0].setUserId("john.smith@yellowfin.com.au");          // must be an existing Yellowfin user

rsr.setPerson(ap);


//Specify which group to exclude from:

AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");                // must be an existing user group

rsr.setGroup(group);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);


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

This function is used to remove a specific Yellowfin user from a specific user group. On doing so, the user will not appear in the group's member list at all.

This request will require the AdministrationPerson object to specify the user, and the AdministrationGroup object to define the user group.

Request Parameters

The following parameters will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "DELUSERFROMGROUP".
Person	Administration Person	Object containing details of the user who is to be deleted from the group. See table below.
Group	Administration Group	Object containing details of the user group. See table below.
OrgRef	String	You may include a Client Org ID to search for this group within a specific client org. If this is not specified, then the group will be searched in the default organization.

These are the main parameters that you need to set in the AdministrationPerson object for this function:

AdministrationPerson Element	Data Type	Description
UserId	String	An existing Yellowfin user to remove them from the group. This value could be a user ID or an email address, depending on the Logon method.

These are the main parameters that you need to set in the AdministrationGroup object for this function:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the group from which the user is to be deleted.

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:remoteAdministrationCall>
         <arg0>
           <loginId>admin@yellowfin.com.au</loginId>
           <password>test</password>
           <orgId>1</orgId>
           <function>DELUSERFROMGROUP</function>
           <person>
                   <userId>binish.sheikh@yellowfin.com.au</userId>
           </person>
           <group>
                   <groupName>Administrators</groupName>
           </group>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

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

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>ed4f6504e415411875b2c359b9384cf9</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Start with a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("DELUSERFROMGROUP");
```

- You may specify a client organization to search for the group within it. But if not included, then the group will be searched in the default (that is, the primary) organization.

```
rsr.setOrgRef("org1");
```

- Set parameters to identify a user who is to be deleted:

```
AdministrationPerson ap = new AdministrationPerson();
ap.setUserId("john.smith@yellowfin.com.au");          // must be an existing Yellowfin user

rsr.setPerson(ap);
```

- Set parameters for the group from which the user is to be deleted:

```
AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");                        //must be an existing user group

rsr.setGroup(group);
```

- Once the request is configured, perform the call:

```
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
```

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: ○ SUCCESS ○ FAILURE

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_deluserfromgroup.jsp.
2. Put the file in the root folder: *Yellowfin/appserver/webapps/ROOT*.

3. Adjust the host, port, admin user, user to remove, and the group name according to your environment.
4. Run *http://<host>:<port>/ws_deluserfromgroup.jsp* from your Internet browser.

```
<%
/*              ws_ deluserfromgroup.jsp              */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number

AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();

AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId("admin@yellowfin.com.au");          // provide your Yellowfin web services admin account
rsr.setPassword("test");                           // set the password of the above account

rsr.setOrgId(1);

rsr.setFunction("DELUSERFROMGROUP");


//Specify the client org (if omitted, the default (primary) org groups will be searched):


rsr.setOrgRef("org1");


//Specify a user to remove from a group:


AdministrationPerson ap = new AdministrationPerson();
ap.setUserId("john.smith@yellowfin.com.au");                 // must be an existing Yellowfin use

rsr.setPerson(ap);


//Specify which group to remove user from:

AdministrationGroup group = new AdministrationGroup();
group.setGroupName("Administrators");                // must be an existing user group

rsr.setGroup(group);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);


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

This function is used to assign a particular dashboard as the default dashboard for a specified user group. Use the ContentResource object to specify the dashboard and the AdministrationGroup object to identify the user group.

Request Parameters

The following parameters will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user ID. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "ASSIGNDEFAULTDASHBOARD".
Group	AdministrationGroup	Object containing details of the user group. See table below.
ContentResources	ContentResource[]	Object array containing the details of the dashboard that is to be made the default one for the group. See table below.

These are the main parameters that you need to set in the AdministrationGroup object for this function:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the group that is to be assigned a default dashboard.
GroupId	Integer	Unique ID of the user group.

These are the main parameters that you need to set in the **ContentResource** object for this function:

ContentResource Element	Data Type	Description
ResourceId	Integer	ID of the dashboard.
ResourceType	String	This must be set to the fixed value "GROUP". Any elements which do not have this value set correctly will be ignored.

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:remoteAdministrationCall>
         <arg0>
                  <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>ASSIGNDEFAULTDASHBOARD</function>
            <group>
                   <groupName>Administrators</groupName>
                   <groupId>11950</groupId>
            </group>
            <contentResources>
                   <resourceId>61251</resourceId>
               <ResourceType>GROUP</ResourceType>
            </contentResources>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE

Response Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>7b5510bf9919823f6067747b5d305984</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Start with a basic request for this function:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("ASSIGNDEFAULTDASHBOARD");
```

- Specify the group by using the AdministrationGroup object:

```
AdministrationGroup administrationGroup = new AdministrationGroup();
administrationGroup.setGroupName("Administrators");
administrationGroup.setGroupId(11950);
```

- Use the ContentResources object to specify a dashboard:

```
ContentResource dashboardContentResource = new ContentResource();
dashboardContentResource.setResourceId(61195);
dashboardContentResource.setResourceType("GROUP");
```

- Then set this object in the request:

```
rsr.setContentResources(new ContentResource[] { dashboardContentResource });
```

- Once the request is configured, perform the call:

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

Initialize the Administration web service. Click here to learn how to do this.

- The response returned will contain the StatusCode parameter. See the Response Parameters 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_assigndefaultdashboard.jsp.
2. Put the file in the root folder: *Yellowfin/appserver/webapps/ROOT*.
3. Adjust the host, port, admin user, user to remove, and the group name according to your environment.
4. Run *http://<host>:<port>/ws_assigndefaultdashboard.jsp* from your Internet browser.

```
<%
/*              ws_assigndefaultdashboard.jsp               */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%

AdministrationServiceResponse rs = null;
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceService ts = new AdministrationServiceServiceLocator("localhost", 8080, "/services
/AdministrationService", false);
AdministrationServiceSoapBindingStub rssbs = (AdministrationServiceSoapBindingStub) ts.
getAdministrationService();

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

// This is the group
AdministrationGroup administrationGroup = new AdministrationGroup();
administrationGroup.setGroupName("Administrators");
administrationGroup.setGroupId(11950);

rsr.setGroup(administrationGroup);

// This is the Dashboard
ContentResource dashboardContentResource = new ContentResource();
dashboardContentResource.setResourceId(61195);
dashboardContentResource.setResourceType("GROUP");

rsr.setContentResources(new ContentResource[] { dashboardContentResource });

rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode())) {
    out.write("Success");
} else {
    out.write("Failure");
    out.write(rs.toString());
}
%>
```