

web

webYellowfin

 LDAPwebLDAPLDAP

YellowfinAdministrationRole

|          |         |                                                       |
|----------|---------|-------------------------------------------------------|
|          |         |                                                       |
| LoginId  | String  | Yellowfin WebIDIDID<br>Yellowfin <b>Yellowfin Web</b> |
| Password | String  |                                                       |
| OrgId    | Integer | YellowfinID1                                          |
| Function | String  | Web <b>LISTROLES</b>                                  |

SOAP

```
<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>
```

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

|            |                                      |                                                                                 |
|------------|--------------------------------------|---------------------------------------------------------------------------------|
| StatusCode | String                               | Web <ul style="list-style-type: none"><li>• SUCCESS</li><li>• FAILURE</li></ul> |
| Roles      | <a href="#">AdministrationRole[]</a> |                                                                                 |

SOAP

```

<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>

```

- 

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- 

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

web

- 

|            |                      |                                                                                 |
|------------|----------------------|---------------------------------------------------------------------------------|
|            |                      |                                                                                 |
| StatusCode | String               | Web <ul style="list-style-type: none"><li>◦ SUCCESS</li><li>◦ FAILURE</li></ul> |
| Roles      | AdministrationRole[] |                                                                                 |

LISTROLES

- 1. **ws\_listroles.jsp**
- 2. rootYellowfin/appserver/webapps/ROOT
- 3.
- 4. **http://<host>:<port>/ws\_listroles.jsp**

```

<%
/*          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());
}
%>

```

AdministrationRoleAdministrationFunctionYellowfinMIREPORTMIREPORTR

|          |         |                                               |
|----------|---------|-----------------------------------------------|
|          |         |                                               |
| LoginId  | String  | Yellowfin WebIDIDID<br>YellowfinYellowfin Web |
| Password | String  |                                               |
| OrgId    | Integer | YellowfinID1                                  |
| Function | String  | WebSAVEROLE                                   |

|      |                    |  |
|------|--------------------|--|
| Role | AdministrationRole |  |
|------|--------------------|--|

AdministrationRole

| AdministrationRole |                        |                                                                                            |
|--------------------|------------------------|--------------------------------------------------------------------------------------------|
| RoleCode           | String                 |                                                                                            |
| RoleName           | String                 | RoleCodeRoleCodeRoleNameTest RoleRoleNameTESTROLERoleCodeTESTROLERoleCodeRoleCodeTESTROLE1 |
| RoleDescription    | String                 |                                                                                            |
| Functions          | AdministrationFunction | SAVEROLE                                                                                   |

webAdministrationFunction

| AdministrationFunction |        |          |
|------------------------|--------|----------|
| FunctionCode           | String | MIREPORT |
| AccessLevelCode        | String | R        |

SOAP

```

<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>

```

|            |        |                                                                                |
|------------|--------|--------------------------------------------------------------------------------|
|            |        |                                                                                |
| StatusCode | String | Web <ul style="list-style-type: none"> <li>SUCCESS</li> <li>FAILURE</li> </ul> |

## SOAP

```
<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>
```

## Java

- 

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- 

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

- 

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



YellowfinOrgRole

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

- SAVEROLE2

```
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");
```



- 

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

- 

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

web

- 

| StatusCode | String | Web                                                                        |
|------------|--------|----------------------------------------------------------------------------|
|            |        | <ul style="list-style-type: none"> <li>SUCCESS</li> <li>FAILURE</li> </ul> |

## SAVEROLE

1. **ws\_saverole.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_saverole.jsp**



```

<%
/*          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());
}
%>

```

#### AdministrationRole

|          |         |                                                       |
|----------|---------|-------------------------------------------------------|
|          |         |                                                       |
| LoginId  | String  | Yellowfin WebIDIDID<br>Yellowfin <b>Yellowfin Web</b> |
| Password | String  |                                                       |
| OrgId    | Integer | YellowfinID1                                          |

|          |                    |               |
|----------|--------------------|---------------|
| Function | String             | WebDELETEROLE |
| Role     | AdministrationRole |               |

#### AdministrationRole

|                           |        |  |
|---------------------------|--------|--|
| <b>AdministrationRole</b> |        |  |
| RoleCode                  | String |  |

#### SOAP

```
<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>
```

|            |        |                                                                                    |
|------------|--------|------------------------------------------------------------------------------------|
|            |        |                                                                                    |
| StatusCode | String | Web <ul style="list-style-type: none"> <li>• SUCCESS</li> <li>• FAILURE</li> </ul> |

#### SOAP

```
<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>
```

Java

- 

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- 

```
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);
```

- 

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

web

- 

| StatusCode | String | Web <ul style="list-style-type: none"><li>◦ SUCCESS</li><li>◦ FAILURE</li></ul> |
|------------|--------|---------------------------------------------------------------------------------|

DELETEROLE

1. **ws\_deleterole.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_deleterole.jsp**

```

<%
/*          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());
}
%>

```

web

LISTGROUPSYellowfinAdministrationGroupID

|          |         |                                                       |
|----------|---------|-------------------------------------------------------|
|          |         |                                                       |
| LoginId  | String  | Yellowfin WebIDIDID<br>Yellowfin <b>Yellowfin Web</b> |
| Password | String  |                                                       |
| OrgId    | Integer | YellowfinID1                                          |
| Function | String  | Web <b>LISTGROUPS</b>                                 |
| OrgRef   | String  | ID                                                    |

SOAP

```
<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>
```

|            |                       |                                                                                 |
|------------|-----------------------|---------------------------------------------------------------------------------|
|            |                       |                                                                                 |
| StatusCode | String                | Web <ul style="list-style-type: none"><li>• SUCCESS</li><li>• FAILURE</li></ul> |
| Groups     | AdministrationGroup[] |                                                                                 |

SOAP

```
<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>
```

## Java

- 

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- ID

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

- 

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

## Web

- 

| StatusCode | String                | Web <ul style="list-style-type: none"><li>◦ SUCCESS</li><li>◦ FAILURE</li></ul> |
|------------|-----------------------|---------------------------------------------------------------------------------|
| Groups     | AdministrationGroup[] |                                                                                 |

- AdministrationGroup.getGroupMembers()AdministrationGroupMembergetGroupMembers()

## LISTGROUPS

1. **ws\_listgroups.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_listgroups.jsp**

```

<%
/*          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());
}
%>

```

|          |         |                                     |
|----------|---------|-------------------------------------|
|          |         |                                     |
| LoginId  | String  | WebIDIDID<br>YellowfinYellowfin Web |
| Password | String  |                                     |
| OrgId    | Integer | YellowfinID1                        |

|          |                                     |             |
|----------|-------------------------------------|-------------|
| Function | String                              | WebGETGROUP |
| Group    | <a href="#">AdministrationGroup</a> |             |
| OrgRef   | String                              | ID          |

AdministrationGroup

|                     |        |  |
|---------------------|--------|--|
| AdministrationGroup |        |  |
| GroupName           | String |  |

SOAP

```
<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>
```

|            |                                       |                                                                             |
|------------|---------------------------------------|-----------------------------------------------------------------------------|
|            |                                       |                                                                             |
| StatusCode | String                                | Web <ul style="list-style-type: none"><li>SUCCESS</li><li>FAILURE</li></ul> |
| Group      | <a href="#">AdministrationGroup[]</a> |                                                                             |

SOAP



```

<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>

```

Java

- 

```

AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("GETGROUP");

```

- ID

```

rsr.setOrgRef("org1");

```

- 

```

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

```

- 

```

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

```

Web

- 

| StatusCode | String                | Web <ul style="list-style-type: none"><li>◦ SUCCESS</li><li>◦ FAILURE</li></ul> |
|------------|-----------------------|---------------------------------------------------------------------------------|
| Group      | AdministrationGroup[] |                                                                                 |

- 

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



YellowfinIpldAdministrationGroupMember.getId()GETUSERBYIPAdministrationPerson

#### GETGROUP

1. **ws\_getgroup.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_getgroup.jsp**

```

<%
/*          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());
}
%>

```

AdministrationPersonAdministrationGroup

| LoginId | String | Yellowfin WebIDIDID<br>Yellowfin <b>Yellowfin Web</b> |
|---------|--------|-------------------------------------------------------|

|            |                      |                                                               |
|------------|----------------------|---------------------------------------------------------------|
| Password   | String               |                                                               |
| OrgId      | Integer              | YellowfinID1                                                  |
| Function   | String               | WebGETUSERGROUPS                                              |
| Person     | AdministrationPerson |                                                               |
| OrgRef     | String               | ID                                                            |
| Parameters | String[]             | AdministrationGroupAdministrationGroupMembertrue<br>TRUEFALSE |

AdministrationPerson

|                      |        |            |
|----------------------|--------|------------|
| AdministrationPerson |        |            |
| UserId               | String | ID<br>IDID |
| IpId                 | String | Ip ID      |

IpId

SOAP

```
<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>GETUSERGROUPS</function>
        <person>
          <!-- Set one of userId and ipId. If both are set, ipId will be used by default. -->
          <userId>admin@yellowfin.com.au</userId>
          <ipId>5</ipId>
        </person>
        <!-- orgRef is optional -->
        <orgRef>org1</orgRef>
        <parameters>
          <!-- TRUE or FALSE. Case insensitive. -->
          <string>true</string>
        </parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>
```

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

|            |                                        |                                                                                 |
|------------|----------------------------------------|---------------------------------------------------------------------------------|
| StatusCode | String                                 | Web <ul style="list-style-type: none"><li>• SUCCESS</li><li>• FAILURE</li></ul> |
| Groups     | <a href="#">AdministrationGroup</a> [] |                                                                                 |

SOAP

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:remoteAdministrationCallResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://webservices.web.mi.hof.com/">
      <remoteAdministrationCallReturn href="#id0"/>
    </ns1:remoteAdministrationCallResponse>
    <multiRef id="id0" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="ns2:AdministrationServiceResponse" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:
ns2="http://service.web.mi.hof.com">
      <binaryAttachments xsi:type="ns2:ArrayOfReportBinaryObject" xsi:nil="true"/>
      <binaryData xsi:type="xsd:string" xsi:nil="true"/>
      <client xsi:type="ns2:AdministrationClientOrg" xsi:nil="true"/>
      <clients xsi:type="ns2:ArrayOfAdministrationClientOrg" xsi:nil="true"/>
      <contentResources xsi:type="ns2:ArrayOfContentResource" xsi:nil="true"/>
      <contentType xsi:type="xsd:string" xsi:nil="true"/>
      <datasources xsi:type="ns2:ArrayOfAdministrationDataSource" xsi:nil="true"/>
      <entityId xsi:type="xsd:int" xsi:nil="true"/>
      <errorCode xsi:type="xsd:int">0</errorCode>
      <fileName xsi:type="xsd:string" xsi:nil="true"/>
      <group xsi:type="ns2:AdministrationGroup" xsi:nil="true"/>
      <groups soapenc:arrayType="ns2:AdministrationGroup[1]" xsi:type="soapenc:Array">
        <groups href="#id1"/></groups>
      <importIssues xsi:type="ns2:ArrayOfImportIssue" xsi:nil="true"/>
      <loadedDataSource xsi:type="ns2:AdministrationDataSource" xsi:nil="true"/>
      <loginSessionId xsi:type="xsd:string" xsi:nil="true"/>
      <messages soapenc:arrayType="soapenc:string[2]" xsi:type="soapenc:Array">
        <messages xsi:type="soapenc:string">Successfully Authenticated User: admin@yellowfin.com.au<
/messages>
        <messages xsi:type="soapenc:string">Web Service Request Complete</messages>
      </messages>
      <parentDashboard xsi:type="ns2:ParentDashboard" xsi:nil="true"/>
      <parentDashboards xsi:type="ns2:ArrayOfParentDashboard" xsi:nil="true"/>
      <parentReportGroups xsi:type="soapenc:Array" xsi:nil="true"/>
      <people xsi:type="ns2:ArrayOfAdministrationPerson" xsi:nil="true"/>
      <person xsi:type="ns2:AdministrationPerson" xsi:nil="true"/>
      <personfavourites xsi:type="ns2:ArrayOfPersonFavourite" xsi:nil="true"/>
      <queryResults xsi:type="ns2:ArrayOfReportRow" xsi:nil="true"/>
      <report xsi:type="ns2:AdministrationReport" xsi:nil="true"/>
      <reportGroups xsi:type="ns2:ArrayOfAdministrationReportGroup" xsi:nil="true"/>
      <reportId xsi:type="xsd:int" xsi:nil="true"/>
      <reports xsi:type="ns2:ArrayOfAdministrationReport" xsi:nil="true"/>
      <roles xsi:type="ns2:ArrayOfAdministrationRole" xsi:nil="true"/>
      <schedule xsi:nil="true"/>
      <schedules xsi:type="ns3:ArrayOfAdministrationSchedule" xsi:nil="true" xmlns:ns3="http://schedule.
service.web.mi.hof.com"/>
      <sessionId xsi:type="xsd:string">974c65a870699f6f4059ba54c15ddc6c</sessionId>
      <statusCode xsi:type="xsd:string">SUCCESS</statusCode>
    </multiRef>
    <multiRef id="id1" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="ns4:AdministrationGroup" xmlns:ns4="http://service.web.mi.hof.com" xmlns:soapenc="http://schemas.
xmlsoap.org/soap/encoding/">
      <groupDescription xsi:type="xsd:string">This group contains all users with the Admin role.<
/groupDescription>
      <groupId xsi:type="xsd:int">11950</groupId>
      <groupInternalReference xsi:type="xsd:string" xsi:nil="true"/>
      <groupMembers soapenc:arrayType="ns4:AdministrationGroupMember[1]" xsi:type="soapenc:Array">
        <groupMembers href="#id2"/></groupMembers>
      <groupName xsi:type="xsd:string">Administrators</groupName>
      <groupStatus xsi:type="xsd:string" xsi:nil="true"/>
    </multiRef>
    <multiRef id="id2" soapenc:root="0" soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="ns5:AdministrationGroupMember" xmlns:ns5="http://service.web.mi.hof.com" xmlns:soapenc="
http://schemas.xmlsoap.org/soap/encoding/">
      <internalId xsi:type="xsd:int">5</internalId>
      <loginId xsi:type="xsd:string">admin@yellowfin.com.au</loginId>
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>

```

Java

- 

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- ID

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

- AdministrationPersonIDip ID

```
AdministrationPerson ap = new AdministrationPerson();
// ap.setUserId("admin@yellowfin.com.au");
ap.setIpId(1);
rsr.setPerson(ap);
```

- 

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

Web

- 

|            |                       |                                                                                 |
|------------|-----------------------|---------------------------------------------------------------------------------|
|            |                       |                                                                                 |
| StatusCode | String                | Web <ul style="list-style-type: none"><li>◦ SUCCESS</li><li>◦ FAILURE</li></ul> |
| Groups     | AdministrationGroup[] |                                                                                 |

- AdministrationGroup.getGroupMembers()AdministrationGroupMembergetGroupMembers()

GETUSERGROUPS

1. **ws\_getusergroups.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_getusergroups.jsp**

```
<%
/*          ws_getusergroups.jsp          */
%>
```

```

<%@ page contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.mi.web.service.AdministrationGroup" %>
<%@ page import="com.hof.mi.web.service.AdministrationGroupMember" %>
<%@ page import="com.hof.mi.web.service.AdministrationPerson" %>
<%@ page import="com.hof.mi.web.service.AdministrationServiceRequest" %>
<%@ page import="com.hof.mi.web.service.AdministrationServiceResponse" %>
<%@ page import="com.hof.mi.web.service.AdministrationServiceService" %>
<%@ page import="com.hof.mi.web.service.AdministrationServiceServiceLocator" %>
<%@ page import="com.hof.mi.web.service.AdministrationServiceSoapBindingStub" %>
<%
    // Adjust host and port number
    AdministrationServiceService adminServiceService = new AdministrationServiceServiceLocator(
        "localhost", 8080,
        "/services/AdministrationService",
        false
    );
    AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub)
adminServiceService.getAdministrationService();
    AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

    rsr.setFunction("GETUSERGROUPS");

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

    // Provide userId or ipId of the user
    AdministrationPerson ap = new AdministrationPerson();
    // Set one of userId and ipId. If both are set, ipId will be used by default.
    // ap.setUserId("admin@yellowfin.com.au");
    ap.setIpId(1);
    rsr.setPerson(ap);

    // Set the first item in the parameters array as "true" or "false" (String, case insensitive) to indicate
    // whether to include AdministrationGroupMember objects in the returned AdministrationGroup objects.
    // If true, the returned groups will contain the information of all their members.
    rsr.setParameters(new String[]{"TRUE"});

    AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

    if ("SUCCESS".equals(rs.getStatusCode())) {
        out.write("Success.<br><br>Groups which the user belongs to:");
        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><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());
        out.write("<br><br>Messages:");
        for (String msg : rs.getMessages()) {
            out.write("<br>");
            out.write(msg);
        }
    }
}

```



%>

IDAdministrationGroupAdministrationGroupMemberYellowfin

|          |                     |                                     |
|----------|---------------------|-------------------------------------|
|          |                     |                                     |
| LoginId  | String              | WebIDIDID<br>YellowfinYellowfin Web |
| Password | String              |                                     |
| OrgId    | Integer             | YellowfinID1                        |
| Function | String              | WebCREATEGROUP                      |
| Group    | AdministrationGroup |                                     |
| OrgRef   | String              | ID                                  |

AdministrationGroup

|                     |                           |  |
|---------------------|---------------------------|--|
| AdministrationGroup |                           |  |
| GroupName           | String                    |  |
| GroupMembers        | AdministrationGroupMember |  |

AdministrationGroupMember

|                           |        |             |
|---------------------------|--------|-------------|
| AdministrationGroupMember |        |             |
| LoginId                   | String | YellowfinID |

SOAP

```
<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>
```

|            |        |                                                                                    |
|------------|--------|------------------------------------------------------------------------------------|
|            |        |                                                                                    |
| StatusCode | String | Web <ul style="list-style-type: none"> <li>• SUCCESS</li> <li>• FAILURE</li> </ul> |

## SOAP

```
<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>
```

## Java

- 

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- ID

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

- 

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

-

```
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);
```

Web

- 

| StatusCode | String | Web <ul style="list-style-type: none"> <li>◦ SUCCESS</li> <li>◦ FAILURE</li> </ul> |
|------------|--------|------------------------------------------------------------------------------------|

#### CREATEGROUP

1. **ws\_creategroup.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_creategroup.jsp**

```

<%
/*          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() );
}
%>

```

|          |         |                                     |
|----------|---------|-------------------------------------|
|          |         |                                     |
| LoginId  | String  | WebIDIDID<br>YellowfinYellowfin Web |
| Password | String  |                                     |
| OrgId    | Integer | YellowfinID1                        |
| Function | String  | WebMODIFYGROUP                      |

|        |                                     |    |
|--------|-------------------------------------|----|
| Group  | <a href="#">AdministrationGroup</a> |    |
| OrgRef | String                              | ID |

AdministrationGroup

|                            |                                           |  |
|----------------------------|-------------------------------------------|--|
| <b>AdministrationGroup</b> |                                           |  |
| GroupName                  | String                                    |  |
| GroupMembers               | <a href="#">AdministrationGroupMember</a> |  |

AdministrationGroupMember

|                                  |        |             |
|----------------------------------|--------|-------------|
| <b>AdministrationGroupMember</b> |        |             |
| LoginId                          | String | YellowfinID |

SOAP

```
<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>
```

|            |        |                                                                         |
|------------|--------|-------------------------------------------------------------------------|
|            |        |                                                                         |
| StatusCode | String | Web                                                                     |
|            |        | <ul style="list-style-type: none"><li>SUCCESS</li><li>FAILURE</li></ul> |

SOAP

```

<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>

```

Java

- 

```

AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("MODIFYGROUP");

```

- ID

```

rsr.setOrgRef("org1");

```

- 

```

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

```

- 

```

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);

```

Web

•

| StatusCode | String | Web <ul style="list-style-type: none"><li>◦ SUCCESS</li><li>◦ FAILURE</li></ul> |
|------------|--------|---------------------------------------------------------------------------------|

#### MODIFYGROUP

1. **ws\_modifygroup.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_modifygroup.jsp**

```

<%
/*          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() );
}
%>

```

AdministrationGroupIDIDYellowfinAccessGroupAccessGroupIdresponse.getGroup().getGroupId()GETGROUP

|          |         |                                     |
|----------|---------|-------------------------------------|
|          |         |                                     |
| LoginId  | String  | WebIDIDID<br>YellowfinYellowfin Web |
| Password | String  |                                     |
| OrgId    | Integer | YellowfinID1                        |
| Function | String  | WebRENANEGROUP                      |



|        |                                     |    |
|--------|-------------------------------------|----|
| Group  | <a href="#">AdministrationGroup</a> |    |
| OrgRef | String                              | ID |

AdministrationGroup

|                            |         |    |
|----------------------------|---------|----|
| <b>AdministrationGroup</b> |         |    |
| GroupId                    | Integer | ID |
| GroupName                  | String  |    |
| GroupDescription           | String  |    |

SOAP

```
<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>
```

|            |        |                                                                             |
|------------|--------|-----------------------------------------------------------------------------|
|            |        |                                                                             |
| StatusCode | String | Web <ul style="list-style-type: none"><li>SUCCESS</li><li>FAILURE</li></ul> |

SOAP

```

<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>

```

Java

- 

```

AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("RENAMEGROUP");

```

- ID

```

rsr.setOrgRef("org1");

```

- 

```

AdministrationGroup group = new AdministrationGroup();

```

- ID

```

group.setGroupId(13002);

```

- 

```

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

rsr.setGroup(group);

```

-

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

Web

- 

| StatusCode | String | Web <ul style="list-style-type: none"><li>◦ SUCCESS</li><li>◦ FAILURE</li></ul> |
|------------|--------|---------------------------------------------------------------------------------|

#### RENAMEGROUP

1. **ws\_renamegroup.jsp**
2. rootYellowfin/appserver/webapps/ROOT
3. ID
4. **http://<host>:<port>/ws\_renamegroup.jsp**

```

<%
/*          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_admin = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);          // adjust host and port number

AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_admin.
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());
}
%>

```

Yellowfinweb

|          |                                     |                                             |
|----------|-------------------------------------|---------------------------------------------|
|          |                                     |                                             |
| LoginId  | String                              | WebIDIDID<br>Yellowfin <b>Yellowfin Web</b> |
| Password | String                              |                                             |
| OrgId    | Integer                             | YellowfinID1                                |
| Function | String                              | Web <b>DELETEGROUP</b>                      |
| Group    | <a href="#">AdministrationGroup</a> |                                             |
| OrgRef   | String                              | ID                                          |

AdministrationGroup

| AdministrationGroup |        |  |
|---------------------|--------|--|
| GroupName           | String |  |

SOAP

```
<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>
```

| StatusCode | String | Web                                                                     |
|------------|--------|-------------------------------------------------------------------------|
|            |        | <ul style="list-style-type: none"><li>SUCCESS</li><li>FAILURE</li></ul> |

SOAP

```
<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>
```

Java

-

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- ID

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

- 

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

rsr.setGroup(group);
```

- 

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

Web

- 

| Status Code | String | Web                                                                            |
|-------------|--------|--------------------------------------------------------------------------------|
|             |        | <ul style="list-style-type: none"> <li>◦ SUCCESS</li> <li>◦ FAILURE</li> </ul> |

## DELETGROUP

1. **ws\_deletgroup.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_deletgroup.jsp**

```

<%
/*          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());
}
%>

```

Yellowfin

AdministrationPersonAdministrationGroup

| LoginId  | String                               | WebIDIDID<br>YellowfinYellowfin Web |
|----------|--------------------------------------|-------------------------------------|
| Password | String                               |                                     |
| OrgId    | Integer                              | YellowfinID1                        |
| Function | String                               | WebINCLUDEUSERINGROUP               |
| Person   | <a href="#">AdministrationPerson</a> |                                     |
| Group    | <a href="#">AdministrationGroup</a>  |                                     |

|        |        |    |
|--------|--------|----|
| OrgRef | String | ID |
|--------|--------|----|

### AdministrationPerson

| AdministrationPerson |        |             |
|----------------------|--------|-------------|
| UserId               | String | YellowfinID |

### AdministrationGroup

| AdministrationGroup |        |  |
|---------------------|--------|--|
| GroupName           | String |  |

### SOAP

```

<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>

```

|            |        |                                                                                |
|------------|--------|--------------------------------------------------------------------------------|
|            |        |                                                                                |
| StatusCode | String | Web <ul style="list-style-type: none"> <li>SUCCESS</li> <li>FAILURE</li> </ul> |

### SOAP



```

<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>

```

Java

- 

```

AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("INCLUDEUSERINGROUP");

```

- 

```

rsr.setOrgRef("org1");

```

- 

```

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

rsr.setPerson(ap);

```

- 

```

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

rsr.setGroup(group);

```

- 

```

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

```

Web

•

| StatusCode | String | Web <ul style="list-style-type: none"><li>◦ SUCCESS</li><li>◦ FAILURE</li></ul> |
|------------|--------|---------------------------------------------------------------------------------|

1. **ws\_includeuseringroup.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_includeuseringroup.jsp**

```

<%
/*          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.*" %>
<%
AdministrationService s_admin = new AdministrationServiceLocator("localhost",8080, "/services
/AdministrationService", false);          // adjust host and port number

AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_admin.
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());
}
%>

```

Yellowfin

AdministrationPersonAdministrationGroup

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

|          |                        |                                     |
|----------|------------------------|-------------------------------------|
| LoginId  | String                 | WebIDIDID<br>YellowfinYellowfin Web |
| Password | String                 |                                     |
| OrgId    | Integer                | YellowfinID1                        |
| Function | String                 | WebINCLUDEUSERSINGROUP              |
| Person   | AdministrationPerson[] |                                     |
| Group    | AdministrationGroup    |                                     |
| OrgRef   | String                 | ID                                  |

AdministrationPerson

|                      |        |             |
|----------------------|--------|-------------|
| AdministrationPerson |        |             |
| UserId               | String | YellowfinID |

AdministrationGroup

|                     |        |  |
|---------------------|--------|--|
| AdministrationGroup |        |  |
| GroupName           | String |  |

SOAP

```
<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>
```

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

|            |        |                                                                                |
|------------|--------|--------------------------------------------------------------------------------|
| StatusCode | String | Web                                                                            |
|            |        | <ul style="list-style-type: none"> <li>• SUCCESS</li> <li>• FAILURE</li> </ul> |

## SOAP

```
<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>
```

## Java

- 

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- 

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

- 

```
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);
```

-

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

rsr.setGroup(group);
```

•

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

Web

•

| StatusCode | String | Web                                                                            |
|------------|--------|--------------------------------------------------------------------------------|
|            |        | <ul style="list-style-type: none"> <li>◦ SUCCESS</li> <li>◦ FAILURE</li> </ul> |

1. **ws\_includeusersingroup.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_includeusersingroup.jsp**

```

<%
/*          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.*" %>
<%
AdministrationService s_adm = new AdministrationServiceLocator("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());
}
%>

```

Yellowfin

| メンバー:                |      |   |
|----------------------|------|---|
| System Administrator | 含む   | × |
| 斎藤 祐佳                | 含めない | × |

AdministrationPersonAdministrationGroup

|          |                      |                                     |
|----------|----------------------|-------------------------------------|
|          |                      |                                     |
| LoginId  | String               | WebIDIDID<br>YellowfinYellowfin Web |
| Password | String               |                                     |
| OrgId    | Integer              | YellowfinID1                        |
| Function | String               | WebEXCLUDEUSERFROMGROUP             |
| Person   | AdministrationPerson |                                     |
| Group    | AdministrationGroup  |                                     |
| OrgRef   | String               | ID                                  |

AdministrationPerson

|                      |        |             |
|----------------------|--------|-------------|
| AdministrationPerson |        |             |
| UserId               | String | YellowfinID |

AdministrationGroup

|                     |        |  |
|---------------------|--------|--|
| AdministrationGroup |        |  |
| GroupName           | String |  |

SOAP

```
<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>
```



|            |        |                                                                                    |
|------------|--------|------------------------------------------------------------------------------------|
|            |        |                                                                                    |
| StatusCode | String | Web <ul style="list-style-type: none"> <li>• SUCCESS</li> <li>• FAILURE</li> </ul> |

## SOAP

```
<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>
```

## Java

•

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

•

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

•

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

rsr.setPerson(ap);
```

•

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

rsr.setGroup(group);
```

- 

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

Web

- 

| StatusCode | String | Web <ul style="list-style-type: none"><li>◦ SUCCESS</li><li>◦ FAILURE</li></ul> |
|------------|--------|---------------------------------------------------------------------------------|

1. **ws\_excludeuserfromgroup.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_excludeuserfromgroup.jsp**

```

<%
/*          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.*" %>
<%
AdministrationService s_adm = new AdministrationServiceLocator("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("EXCLUDEUSERFROMMGROUP");

//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());
}
%>

```

Yellowfin

| メンバー:                |      |   |
|----------------------|------|---|
| System Administrator | 含む   | × |
| 斎藤 祐佳                | 含めない | × |



AdministrationPersonAdministrationGroup

|          |                        |                                     |
|----------|------------------------|-------------------------------------|
|          |                        |                                     |
| LoginId  | String                 | WebIDIDID<br>YellowfinYellowfin Web |
| Password | String                 |                                     |
| OrgId    | Integer                | YellowfinID1                        |
| Function | String                 | WebEXCLUDEUSERSFROMGROUP            |
| Person   | AdministrationPerson[] |                                     |
| Group    | AdministrationGroup    |                                     |
| OrgRef   | String                 | ID                                  |

AdministrationPerson

|                      |        |             |
|----------------------|--------|-------------|
| AdministrationPerson |        |             |
| UserId               | String | YellowfinID |

AdministrationGroup

|                     |        |  |
|---------------------|--------|--|
| AdministrationGroup |        |  |
| GroupName           | String |  |

|            |        |                                                                                |
|------------|--------|--------------------------------------------------------------------------------|
|            |        |                                                                                |
| StatusCode | String | Web<br><ul style="list-style-type: none"><li>SUCCESS</li><li>FAILURE</li></ul> |

Java

- 

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- 

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

- 

```
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);
```

- 

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

rsr.setGroup(group);
```

- 

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

Web

- 

| StatusCode | String | Web <ul style="list-style-type: none"><li>◦ SUCCESS</li><li>◦ FAILURE</li></ul> |
|------------|--------|---------------------------------------------------------------------------------|

1. **ws\_excludeusersfromgroup.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_excludeusersfromgroup.jsp**

```

<%
/*          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("EXCLUDEUSERFROMMGROUP");

//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());
}
%>

```

Yellowfin

AdministrationPersonAdministrationGroup

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

|          |                                      |                                             |
|----------|--------------------------------------|---------------------------------------------|
| LoginId  | String                               | WebIDIDID<br>Yellowfin <b>Yellowfin Web</b> |
| Password | String                               |                                             |
| OrgId    | Integer                              | YellowfinID1                                |
| Function | String                               | Web <b>DELUSERFROMGROUP</b>                 |
| Person   | <a href="#">AdministrationPerson</a> |                                             |
| Group    | <a href="#">AdministrationGroup</a>  |                                             |
| OrgRef   | String                               | ID                                          |

#### AdministrationPerson

|                             |        |             |
|-----------------------------|--------|-------------|
| <b>AdministrationPerson</b> |        |             |
| UserId                      | String | YellowfinID |

#### AdministrationGroup

|                            |        |  |
|----------------------------|--------|--|
| <b>AdministrationGroup</b> |        |  |
| GroupName                  | String |  |

#### SOAP

```
<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>
```

|            |        |                                                                                    |
|------------|--------|------------------------------------------------------------------------------------|
|            |        |                                                                                    |
| StatusCode | String | Web <ul style="list-style-type: none"> <li>• SUCCESS</li> <li>• FAILURE</li> </ul> |

## SOAP

```
<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>
```

## Java

- 

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- 

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

- 

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

rsr.setPerson(ap);
```

- 

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

rsr.setGroup(group);
```

- 

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

## Web



•

| StatusCode | String | Web <ul style="list-style-type: none"><li>◦ SUCCESS</li><li>◦ FAILURE</li></ul> |
|------------|--------|---------------------------------------------------------------------------------|

1. **ws\_deluserfromgroup.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_deluserfromgroup.jsp**

```

<%
/*          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("DELUSERFROMMGROUP");

//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());
}
%>

```

ContentResourceAdministrationGroup

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

|                 |                     |                                     |
|-----------------|---------------------|-------------------------------------|
| LoginId         | String              | WebIDIDID<br>YellowfinYellowfin Web |
| Password        | String              |                                     |
| OrgId           | Integer             | YellowfinID1                        |
| Function        | String              | WebASSIGNDEFAULTDASHBOARD           |
| Group           | AdministrationGroup |                                     |
| ContentResource | ContentResource[]   |                                     |

AdministrationGroup

|                     |         |    |
|---------------------|---------|----|
| AdministrationGroup |         |    |
| GroupName           | String  |    |
| GroupId             | Integer | ID |

ContentResource

|                 |         |       |
|-----------------|---------|-------|
| ContentResource |         |       |
| ResourceId      | Integer | ID    |
| ResourceType    | String  | GROUP |

SOAP

```
<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>
```

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

|            |        |                                                                                    |
|------------|--------|------------------------------------------------------------------------------------|
| StatusCode | String | Web <ul style="list-style-type: none"> <li>• SUCCESS</li> <li>• FAILURE</li> </ul> |
|------------|--------|------------------------------------------------------------------------------------|

## SOAP

```
<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>
```

## Java

- 

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- AdministrationGroup

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

- ContentResource

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

- 

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

-

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

Web

- StatusCode

1. **ws\_assigndefaultdashboard.jsp**
2. rootYellowfin/appserver/webapps/ROOT
- 3.
4. **http://<host>:<port>/ws\_assigndefaultdashboard.jsp**

```
<%
/*          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());
}
%>
```

