User Replication & Management Services

User replication involves synchronising each user in the third-party application with a named user in Yellowfin. This allows Yellowfin to identify the user who is logged in, and to apply any restrictions that may be required. Synchronization is usually performed using web service calls from the third-party application to Yellowfin. This can also be managed manually if users in the third-party application are generally static.

This section will outline how to create, manipulate, and delete users via web services. It is assumed that the web service is called to mirror user changes immediately after a user modification is made in the third-party application.

Main User Management Functions

This function creates a new user account in Yellowfin.

Request Parameters

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user Id. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "ADDUSER".
Person	AdministrationPerson	Object containing all of the new user’s details for the user creation process. See table below.
OrgRef	String	Client Org Internal Reference Id (optional). This will create a new user in the referenced Client Org. If this is not set, then the new user will be created in default (primary) org.

These are the mandatory parameters that you need to set in the AdministrationPerson object to create a new user:

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the new user. This can be the user ID or the email address, depending on the Logon ID method.
Password	String	Password of the new user. This must comply with Yellowfin's Password Policy.
FirstName	String	The first name of the new user.
LastName	String	The last name of the new user.
RoleCode	String	Set the user role for this new user. For example, YFREPORTCONSUMER. **Note:** You can get a list of Yellowfin's role codes from the configured database, for example, by using a SQL query: SELECT * FROM OrgRole
EmailAddress	String	Email address of the new user.

Request Example

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

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.
mi.hof.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <web:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>ADDUSER</function>
            <person>
                    <userId>binish.sheikh@yellowfin.com.au</userId>
                <emailAddress>binish.sheikh@yellowfin.com.au</emailAddress>
                <password>admin</password>
                <firstName>Binish</firstName>
                     <lastName>Sheikh</lastName>
                <roleCode>YFREPORTCONSUMER</roleCode>
            </person>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Reponse Parameters

The response returned will contain these parameters:

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

Reponse Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Creating New User via Webservices...</messages>
            <messages>Creating SessionBean for webservices user...</messages>
            <messages>Looking up user...</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>8090cb7879f7d5e72eab9625772c00b5</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

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

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();


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

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

- If you need to create a new user in a specific client organization, add this to your code

```
rsr.setOrgRef("org1");      // A new user will be added to the client org with "org1" as an organization
reference ID
```

  If you do not define the orgRef parameter, the new user will be created in the default (primary) organization.

- The ADDUSER function requires AdministrationPerson object where you define the new Yellowfin user details:

```
AdministrationPerson ap = new AdministrationPerson();
```

- To create a new user, you need to fill these mandatory parameters: UserId, FirstName, LastName, RoleCode, Password, EmailAddress:

```
ap.setUserId("john.smith@yellowfin.com.au");               // if Yellowfin authentication option is set
to "email address"
ap.setFirstName("John");
ap.setLastName("Smith");
ap.setRoleCode("YFREPORTCONSUMER");               // Yellowfin role codes can be found performing this
query against
                                                                                                    //
Yellowfin configuration database: SELECT * FROM OrgRole

ap.setPassword("test");                                      // Password must comply with your
Yellowfin password policy
ap.setEmailAddress("john.smith@yellowfin.com.au");
```

  Other parameters of the AdministrationPerson object are optional.

- Pass the 'ap' object to the request:

```
rsr.setPerson(ap);
```

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

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

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

- The response returned will contain the StatusCode parameter. See the Response Parameters table above for details.

Complete Example

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

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

```
<%
/*              ws_adduser.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 service admin account
rsr.setPassword("test");                           // change this to the password of the above account
rsr.setOrgId(1);
rsr.setFunction("ADDUSER");


AdministrationPerson ap = new AdministrationPerson();


ap.setUserId("john.smith@yellowfin.com.au");                // If Yellowfin authentication option is set to
"email address"
ap.setFirstName("John");
ap.setLastName("Smith");
ap.setRoleCode("YFREPORTCONSUMER");                // Yellowfin role codes can be found performing this query
against
                                                                                                    // Yellowfin
configuration database: SELECT * FROM OrgRole

ap.setPassword("test");                                 // Password must comply with your Yellowfin password
policy
ap.setEmailAddress("john.smith@yellowfin.com.au");
rsr.setPerson(ap);


AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

This function creates users in bulk. It is similar to the ADDUSER function, however this requires that you pass an *array* of AdministrationPerson objects.

Request Parameters

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user Id. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "ADDUSERS".
Person	Administration Person[]	Object array containing details of the new users to be added in bulk. See table below.
OrgRef	String	Client Org Internal Reference Id (optional). This will create the new users in the referenced Client Org. If this is not set, then the users will be created in default (primary) org.

These are the mandatory parameters that you need to set in the **AdministrationPerson** object to create a new user:

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the new user. This can be the user ID or the email address, depending on the Logon ID method.
Password	String	Password of the new user. This must comply with Yellowfin's Password Policy.
FirstName	String	The first name of the new user.
LastName	String	The last name of the new user.
RoleCode	String	Set the user role for this new user. For example, YFREPORTCONSUMER. Note: You can get a list of Yellowfin's role codes from the configured database, for example, by using a SQL query: SELECT * FROM OrgRole
EmailAddress	String	Email address of the new user.

Reponse Parameters

The response returned will contain these parameters:

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

Instructions

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

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

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();


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

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

- If you need to create new users in a specific client organization, add this to your code:

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

If you do not define the orgRef parameter, the new users will be created in the default (primary) organization.

- Use AdministrationPerson object to define the new Yellowfin users details:

```
AdministrationPerson[] ap = new AdministrationPerson[2];
```

- Add details of the first user by providing these mandatory parameters: UserId, FirstName, LastName, RoleCode, Password, EmailAddress:

```
ap[0] = new AdministrationPerson();
ap[0].setUserId("user1@yellowfin.com.au");
ap[0].setFirstName("user1");
ap[0].setLastName("Lastname1");
ap[0].setRoleCode("YFREPORTCONSUMER");
ap[0].setPassword("test");
ap[0].setEmailAddress("user1@yellowfin.com.au");
```

Other parameters of the AdministrationPerson object are optional.

- Similary, provide the mandatory details of the other user:

```
ap[1] = new AdministrationPerson();
ap[1].setUserId("user2@yellowfin.com.au");
ap[1].setFirstName("user2");
ap[1].setLastName("Lastname2");
ap[1].setRoleCode("YFREPORTCONSUMER");
ap[1].setPassword("test");
ap[1].setEmailAddress("user2@yellowfin.com.au");
```

This can be repeated for as many users as required.

- Pass the 'ap' object to the request:

```
rsr.setPeople(ap);
```

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

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

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

- The response returned will contain the StatusCode parameter. See the Response Parameters table above for details.

Complete Example

Below is a complete example of the ADDUSERS function. This example code adds two new Yellowfin users, user1@yellowfin.com.au and user2@yellowfin.com.au, in the default organization.

To use it for yourself, carry out the following the steps:

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

```
<%
/*              ws_addusers.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("ADDUSERS");

AdministrationPerson[] ap = new AdministrationPerson[2];

ap[0] = new AdministrationPerson();
ap[0].setUserId("user1@yellowfin.com.au");
ap[0].setFirstName("user1");
ap[0].setLastName("Lastname1");
ap[0].setRoleCode("YFREPORTCONSUMER");
ap[0].setPassword("test");
ap[0].setEmailAddress("user1@yellowfin.com.au");

ap[1] = new AdministrationPerson();
ap[1].setUserId("user2@yellowfin.com.au");
ap[1].setFirstName("user2");
ap[1].setLastName("Lastname2");
ap[1].setRoleCode("YFREPORTCONSUMER");
ap[1].setPassword("test");
ap[1].setEmailAddress("user2@yellowfin.com.au");


rsr.setPeople(ap);


AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

This function allows multiple users to be created, without adding duplicates. It works similarly to the ADDUSERS function, however in this case, if the login ID or email of a potential new user is already in use, or the password isn't supplied, then a 'no exceptions' error will be generated and the user will not be created. The response will contain an array of AdministrationPerson objects with failed users that were not added.

Request Parameters

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user Id. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "ADDUSERSIGNOREDUPLICATES".
Person	Administration Person[]	Object array containing details of the new users to be added and validated for duplicates. See table below.
OrgRef	String	Client Org Internal Reference Id (optional). This will create the new users in the referenced Client Org. If this is not set, then the users will be created in default (primary) org.

These are the mandatory parameters that you need to set in the **AdministrationPerson** object to create a new user:

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the new user. This can be the user ID or the email address, depending on the Logon ID method.
Password	String	Password of the new user. This must comply with Yellowfin's Password Policy.
FirstName	String	The first name of the new user.
LastName	String	The last name of the new user.
RoleCode	String	Set the user role for this new user. For example, YFREPORTCONSUMER. Note: You can get a list of Yellowfin's role codes from the configured database, for example, by using a SQL query: SELECT * FROM OrgRole
EmailAddress	String	Email address of the new user.

Reponse Parameters

The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
People	AdministrationPerson[]	Failed users whose accounts were not created.

Instructions

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

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

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();


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

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

- If you need to create new users in a specific client organization, add this to your code:

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

  If you do not define the orgRef parameter, the new users will be created in the default (primary) organization.

- Use AdministrationPerson object to define the new Yellowfin users details:

```
AdministrationPerson[] ap = new AdministrationPerson[2];
```

- Add details of the first user by providing these mandatory parameters: UserId, FirstName, LastName, RoleCode, Password, EmailAddress:

```
ap[0] = new AdministrationPerson();
ap[0].setUserId("user1@yellowfin.com.au");
ap[0].setFirstName("user1");
ap[0].setLastName("Lastname1");
ap[0].setRoleCode("YFREPORTCONSUMER");
ap[0].setPassword("test");
ap[0].setEmailAddress("user1@yellowfin.com.au");
```

  Other parameters of the AdministrationPerson object are optional.

- Similary, provide the mandatory details of the other user:

```
ap[1] = new AdministrationPerson();
ap[1].setUserId("user2@yellowfin.com.au");
ap[1].setFirstName("user2");
ap[1].setLastName("Lastname2");
ap[1].setRoleCode("YFREPORTCONSUMER");
ap[1].setPassword("test");
ap[1].setEmailAddress("user2@yellowfin.com.au");
```

  This can be repeated for as many users as required.

- Pass the 'ap' object to the request:

```
rsr.setPeople(ap);
```

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

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

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

- The response returned will contain the StatusCode and People parameters. See the Response Parameters table above for details.

- Use this command to retrieve failed users:

```
AdministrationPerson[] failed_users = rs.getPeople();
```

This function will delete a specified user from Yellowfin. **Note:** To remove a user from a client organization, you should perform the REMOVEUSERACCESS call.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user Id. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "DELETEUSER" or "DELUSER".
Person	AdministrationPerson	Object containing details of the user account that needs to be deleted. Note: See table below.
OrgRef	String	Client Org Internal Reference Id (optional). This will log the user into a given Client Org. If this is not set, then the user will be prompted with the Client Org Selection page on login.

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

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the user that you wish to delete. This user should already exist in Yellowfin. This value could be a user ID or an email address, depending on the Logon ID method.

Request Example

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

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.
mi.hof.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <web:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>DELETEUSER</function>
            <person>
                    <userId>binish.sheikh@yellowfin.com.au</userId>
            </person>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Reponse Elements

The response returned will contain these parameters:

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

Response Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2: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>cc8c2b81d06485fdd20fe4cd24f5f70c</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Here is a basic request for this call, which includes logging in as the admin user and specifying the web service call to perform.

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- The function requires an AdministrationPerson object, which is used to specify which user to delete, by providing their ID (for example, their email address or another type of ID depending on the Login ID method):

```
AdministrationPerson ap = new AdministrationPerson();

ap.setUserId("test@yellowfin.com.au");                  // test@yellowfin.com.au should be an existing
Yellowfin user.
rsr.setPerson(ap);
```

- After configuring the request, perform the call:

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

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

- The response will contain the StatusCode parameter. See the Response Parameters table above for more detail.

Complete Example

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

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

```
<%
/*              ws_deleteuser.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("DELETEUSER");


AdministrationPerson ap = new AdministrationPerson();

ap.setUserId("test@yellowfin.com.au");
rsr.setPerson(ap);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);


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

Retrieving & Validating User Information

Once a user has been created, the user's details can be retrieved using a web service call. The User ID field in the AdministrationPerson object is used to identify the user. As a result, a populated AdministrationPerson object will be returned. For security reasons, passwords will not be returned and will be NULL. User information can also be validated against the application in this section.

This function retrieves an existing Yellowfin user's details. It accepts AdministrationPerson as a parameter which you can use to identify the user. The response will contain the AdministrationPerson object with full user details.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description

LoginId	String	The ID of a web services admin user who logs in to perform this function. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "GETUSER".
Person	AdministrationPerson	Object containing details of the user whose information is to be retrieved. Note: See table below.
OrgRef	String	Client Org Internal Reference Id (optional). This will log the user into a given Client Org. If this is not set, then the user will be prompted with the Client Org Selection page on login.

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

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the user whose information is to be retrieved. This can be the user ID or the email address, depending on the Logon ID method.

Request Example

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

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

Reponse Elements

The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
Person	AdministrationPerson	Object with the user details

Reponse Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Getting user information...</messages>
            <messages>Getting user information...</messages>
            <messages>Web Service Request Complete</messages>
            <person>
               <emailAddress>admin@yellowfin.com.au</emailAddress>
               <firstName>System</firstName>
               <initial/>
               <ipId>5</ipId>
               <languageCode>EN</languageCode>
               <lastName>Administrator</lastName>
               <roleCode>YFADMIN</roleCode>
               <salutationCode/>
               <status>ACTIVE</status>
               <timeZoneCode>AUSTRALIA/SYDNEY</timeZoneCode>
               <userId>admin@yellowfin.com.au</userId>
            </person>
            <sessionId>70dc3c7158a340e19b590f0ed6ea6a8b</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Following is a basic request for this call, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- Now provide the user ID via the AdministrationPerson object:

```
AdministrationPerson ap = new AdministrationPerson();
ap.setUserId("john.smith@yellowfin.com.au");
rsr.setPerson(ap);
```

- Once the request is configured, perform the call:

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

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

- The response will contain the StatusCode and Person parameters. See the Response Parameters table for details.
- To display the retrieved user details, use the following example:

```
if ("SUCCESS".equals(rs.getStatusCode()) ) {
        ap = rs.getPerson();
        out.write("UserId:" + ap.getUserId() + "<br>");
        out.write("Password:" + ap.getPassword() + "<br>");
        out.write("FirstName:" + ap.getFirstName() + "<br>");
        out.write("LastName:" + ap.getLastName() + "<br>");
        out.write("Initial:" + ap.getInitial() + "<br>");
        out.write("SalutationCode:" + ap.getSalutationCode() + "<br>");
        out.write("RoleCode:" + ap.getRoleCode() + "<br>");
        out.write("EmailAddress:" + ap.getEmailAddress() + "<br>");
        out.write("LanguageCode:" + ap.getLanguageCode() + "<br>");
        out.write("IpId:" + ap.getIpId() + "<br>");
        out.write("TimeZoneCode:" + ap.getTimeZoneCode() + "<br>");
        out.write("Status:" + ap.getStatus() + "<br>");
} else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode());
}
```

Complete Example

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

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

```
<%
/*              ws_getuser.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("GETUSER");

AdministrationPerson ap = new AdministrationPerson();
ap.setUserId("john.smith@yellowfin.com.au");
rsr.setPerson(ap);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
        ap = rs.getPerson();
        out.write("User Id:" + ap.getUserId() + "<br>");
        out.write("Password:" + ap.getPassword() + "<br>");
        out.write("First Name:" + ap.getFirstName() + "<br>");
        out.write("Last Name:" + ap.getLastName() + "<br>");
        out.write("Initial:" + ap.getInitial() + "<br>");
        out.write("Salutation Code:" + ap.getSalutationCode() + "<br>");
        out.write("Role Code:" + ap.getRoleCode() + "<br>");
        out.write("Email Address:" + ap.getEmailAddress() + "<br>");
        out.write("Language Code:" + ap.getLanguageCode() + "<br>");
        out.write("IpId:" + ap.getIpId() + "<br>");
        out.write("Time Zone Code:" + ap.getTimeZoneCode() + "<br>");
        out.write("Status:" + ap.getStatus() + "<br>");
} else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode() );
}
%>
```

This functions retrieves details of all the users in a specified client organization. The information is retrieved in an array of AdministrationPerson objects. If a client organization is not specified, then all the users will be retrieved. You can use the setParameters() method to specify a searching criteria for users being retrieved. For security reasons, passwords will not be returned and will be NULL.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description

LoginId	String	The ID of a web services admin user who logs in to perform this function. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "GETUSER".
OrgRef	String	Client Org Internal Reference Id (optional). Specify a client organization, otherwise all the client organizations will get searched.

Request Example

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

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

Reponse Elements

The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
People	AdministrationPerson[]	Array of objects with the users’ details.

Response Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <people>
               <firstName>System</firstName>
               <ipId>5</ipId>
               <lastName>Administrator</lastName>
               <userId>admin@yellowfin.com.au</userId>
            </people>
            <sessionId>2c32528279baa26b730f9e3c8787880d</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Here's an example of a request to retrieve all users, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- Specify a client organization, otherwise all the client organizations will get searched:

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

- Searching Criteria: The GETALLUSERS function accepts an array of 2 Strings. The first string (searchingCriteria[0]) will be compared with Yellowfin database users' first names, last names, email left or email right, using the condition LIKE '%John%'. The second string (searchingCriteria[1]) will be compared with email right (domain) of the Yellowfin database users.

```
String[] searchingCriteria = new String[] {"John","yellowfin.com.au"};
rsr.setParameters(searchingCriteria);
```

- Once the request is configured, perform the call:

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

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

- The response will contain the StatusCode and People parameters. See the Response Parameters table above for more details.

Complete Example

Below is a complete Java example of the GETALLUSERS function. To use it for yourself, carry out the following the steps:

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

```
<%
/*              ws_getallusers.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");
rsr.setPassword("test");
rsr.setOrgId(1);

rsr.setFunction("GETALLUSERS");

rsr.setOrgRef("org1");

String[] searchingCriteria = new String[] {"John","yellowfin.com.au"};
rsr.setParameters(searchingCriteria);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

This function will retrieve details of a user by their IP. It accepts the AdministrationPerson object as a parameter, which can be used to identify the user, by providing their IPID (in the Yellowfin configuration database, this is the **IpId** field of the **Person** table). The response will contain the AdministrationPerson object with the full details of the user.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	The ID of a web services admin user who logs in to perform this function. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "GETUSERBYIP".
Person	Administration Person	Object containing details of the user whose information is to be retrieved. **Note:** See table below.

Following is the main parameter that you need to set in the AdministrationPerson object for this function:

AdministrationPerson Element	Data Type	Description
IpId	Integer	IP ID of the user whose information is to be retrieved.

Request Example

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

```
 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.
mi.hof.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <web:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>GETUSERBYIP</function>
                  <person>
                <ipId>5</ipId>
            </person>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Reponse Elements

The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
Person	AdministrationPerson	Object with the user details

Reponse Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Getting user information...</messages>
            <messages>Getting user information...</messages>
            <messages>Web Service Request Complete</messages>
            <person>
               <emailAddress>admin@yellowfin.com.au</emailAddress>
               <firstName>System</firstName>
               <initial/>
               <ipId>5</ipId>
               <languageCode>EN</languageCode>
               <lastName>Administrator</lastName>
               <roleCode>YFADMIN</roleCode>
               <salutationCode/>
               <status>ACTIVE</status>
               <timeZoneCode>AUSTRALIA/SYDNEY</timeZoneCode>
               <userId>admin@yellowfin.com.au</userId>
            </person>
            <sessionId>81e76f1222d0dd12d9871efc7cbf0811</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

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

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- Then use the AdministrationPerson object to identify the user whose details are to be retrieved, by providing their IpId:

```
AdministrationPerson ap = new AdministrationPerson();

ap.setIpId(5);                  //IpId of the admin@yellowfin.com.au account
rsr.setPerson(ap);
```

- Once the request is configured, perform the call:

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

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

- This call's response will contain thes StatusCode and Person parameters. See the Response Parameters table above for more details.
- Use this example to display the result of this call:

```
if ("SUCCESS".equals(rs.getStatusCode()) ) {
        ap = rs.getPerson();
        out.write("UserId: " + ap.getUserId() + "<br>");
        out.write("Password: " + ap.getPassword() + "<br>");
        out.write("FirstName: " + ap.getFirstName() + "<br>");
        out.write("LastName: " + ap.getLastName() + "<br>");
        out.write("Initial: " + ap.getInitial() + "<br>");
        out.write("SalutationCode: " + ap.getSalutationCode() + "<br>");
        out.write("RoleCode: " + ap.getRoleCode() + "<br>");
        out.write("EmailAddress: " + ap.getEmailAddress() + "<br>");
        out.write("LanguageCode: " + ap.getLanguageCode() + "<br>");
        out.write("IpId: " + ap.getIpId() + "<br>");
        out.write("TimeZoneCode: " + ap.getTimeZoneCode() + "<br>");
        out.write("Status: " + ap.getStatus() + "<br>");
} else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode());
}
```

Complete Example

Below is a complete example of the GETUSERBYIP function. To use it for yourself, first carry out the following the steps:

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

```
<%
/*              ws_getuserbyip.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("GETUSERBYIP");

AdministrationPerson ap = new AdministrationPerson();

ap.setIpId(5);
rsr.setPerson(ap);
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
        ap = rs.getPerson();
        out.write("User Id: " + ap.getUserId() + "<br>");
        out.write("Password: " + ap.getPassword() + "<br>");
        out.write("First Name: " + ap.getFirstName() + "<br>");
        out.write("Last Name: " + ap.getLastName() + "<br>");
        out.write("Initial: " + ap.getInitial() + "<br>");
        out.write("Salutation Code: " + ap.getSalutationCode() + "<br>");
        out.write("Role Code: " + ap.getRoleCode() + "<br>");
        out.write("Email Address: " + ap.getEmailAddress() + "<br>");
        out.write("Language Code: " + ap.getLanguageCode() + "<br>");
        out.write("IpId: " + ap.getIpId() + "<br>");
        out.write("Time Zone Code: " + ap.getTimeZoneCode() + "<br>");
        out.write("Status: " + ap.getStatus() + "<br>");
} else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode() );
}
%>
```

This web service is used to retrieve users' avatar images. If an avatar image is not set up for a user in Yellowfin, then no image will be retrieved for that user. This service requires the users' internal IDs (that is, their IP ID).

Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description

LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method. This account must have the “web services” role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "GETAVATARS".
Parameters	String[]	An array containing the internal IDs (or IpId) of Yellowfin users' whose avatars you want to retrieve. Each ID must be passed as a String, even though it's an Integer.

Request Example

Below is a SOAP XML example for this request:

```
<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>GETAVATARS</function>
            <parameters>5</parameters>
         </arg0>
          </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

The returned response will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
binaryAttachments	ReportBinaryObject[]	Contains avatar images for the specified users.

Response Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <binaryAttachments>
               <data>iVBORw0KGgoAAAANSUhEUgAAA ... your image string</data>
            </binaryAttachments>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>5547ebe153fd0fc7fcf63014ffe61b5c</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Start with a basic request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- You can specify the users' IP IDs:

```
rsr.setParameters(new String[] {"5","13073"});
```

- Once the request is configured, perform the call:

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

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

- The response will contain a StatusCode element, along with a BinaryAttachment array.
- To get the first image, use the following:

```
byte[] data = response.getBinaryAttachments[0].getData();
```

Complete Example

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

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

```
<%
/*              ws_getavatars.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.*" %>
<html>
<body>
<%
AdministrationServiceService s_adm = new AdministrationServiceServiceLocator("localhost",8080, "/services
/AdministrationService", false);        // adjust host and port number
AdministrationServiceSoapBindingStub adminService = (AdministrationServiceSoapBindingStub) s_adm.
getAdministrationService();
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

rsr.setFunction("GETAVATARS");
rsr.setParameters(new String[] {"5","13073"});
AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
        ReportBinaryObject[] objects = rs.getBinaryAttachments();
        if (objects != null) {
                byte[] data,encodeBase64;
                String base64Encoded;

                for(ReportBinaryObject o: objects){
                        data = o.getData();
                        if (data != null){

                                encodeBase64 = java.util.Base64.getEncoder().encode(data);
                                base64Encoded = new String(encodeBase64, "UTF-8");
                                if (base64Encoded != null)
                                        out.write("<br>");
                                %>
                                <img src="<%=base64Encoded%>" alt="No image">
                                <%
                        }
                }
        }
} else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode());
}
%>
</body>
</html>
```

This function validates a Yellowfin user. You can specify the user by using the AdministrationPerson as a parameter.

The response will be SUCCESS if the user with provided details exists. Otherwise, the response will return code 25 (COULD_NOT_AUTHENTICATE_USER) if the user is not valid.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	The ID of a web services admin user who logs in to perform this function. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "VALIDATEUSER".
Person	Administration Person	Object containing details of the user who is to be validated. **Note:** See table below.
OrgRef	String	Client organization reference ID (optional). Provide this if you need to validate the user of a specific organization.

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

AdministrationPerson Element	Data Type	Description
User ID	String	User ID of the user to validate. This can be the user ID or the email address, depending on the Logon ID method.
Password	String	Password of the above account.

Request Example

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

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

Reponse Elements

The response returned will contain these parameters:

Response Element	Data Type	Description

StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
Person	AdministrationPerson	Object with the user details.

Reponse Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <person>
               <ipId>5</ipId>
               <userId>admin@yellowfin.com.au</userId>
            </person>
            <sessionId>0ee8ae23d21ff01f365d1e03ea49c47a</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

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

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- Now you need to provide user ID via the AdministrationPerson object:

```
AdministrationPerson ap = new AdministrationPerson();

ap.setUserId("john.smith@yellowfin.com.au");
ap.setPassword("test");

rsr.setPerson(ap);
```

- You can provide a client organization reference ID if you need to validate the user of a specific organization:

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

If the OrgRef parameter is omitted, the user will be validated against the default (primary) organization.

- Once the request is configured, perform the call:

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

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

- The response will contain the StatusCode parameter.

Complete Example

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

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

```
<%
/*              ws_validateuser.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("VALIDATEUSER");

AdministrationPerson ap = new AdministrationPerson();

ap.setUserId("john.smith@yellowfin.com.au");
ap.setPassword("test");

rsr.setPerson(ap);
rsr.setOrgRef("org1");

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

This call validates a user's password. If the password is expired, the call will return SUCCESS as the StatusCode, otherwise it will return FAILURE.

Keep in mind that for a freshly created user who has not yet logged into Yellowfin, the call retrieves SUCCESS meaning that their password is expired. This happens because for every new user, Yellowfin is required to reset the password at the first login.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user Id. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "VALIDATEPASSWORD".
Person	AdministrationPerson	Object containing details of user whose password is to be validated. See table below.

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

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the user whose password is to be validated. This can be the user ID or the email address, depending on the Logon ID method.
Password	String	Password of the above account.

Request Example

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

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

Reponse Elements

The response returned will contain these parameters:

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

Reponse Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>fb024963bdc94034557f95b01f53a138</sessionId>
            <statusCode>FAILURE</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

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

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- Then provide the user ID via the AdministrationPerson object:

```
AdministrationPerson ap = new AdministrationPerson();
ap.setUserId("john.smith@yellowfin.com.au");
ap.setPassword("test");

rsr.setPerson(ap);
```

- Once the request is configured, perform the call:

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

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

- The response will contain the StatusCode parameter. See the Response Parameters table above for more details.

Complete Example

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

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

```
<%
/*              ws_validatepwd.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("VALIDATEPASSWORD");

AdministrationPerson ap = new AdministrationPerson();

ap.setUserId("john.smith@yellowfin.com.au");
ap.setPassword("test");
rsr.setPerson(ap);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
if ("SUCCESS".equals(rs.getStatusCode()) ) {
        out.write("Password is expired");
} else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode() );
}
%>
```

Manipulating User Information

A user's details can be modified at a later time using a web service call. The User ID field in the AdministrationPerson object is used to identify the user, so this cannot be changed. The rest of the fields within an AdministrationPerson object are populated with the new changes. .

This function will change the password for the specified Yellowfin user. The password will be reset through the AdministrationPerson parameter, based on their user ID (email address or another type of ID depending on the Login ID method).

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user Id. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "CHANGEPASSWORD".
Person	AdministrationPerson	Object containing details of the user whose password is to be changed. See table below.

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

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the user whose password is to be changed. This can be the user ID or the email address, depending on the Logon ID method.
Password	String	The new password of the user. This must comply with Yellowfin's Password Policy.

Request Example

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

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.
mi.hof.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <web:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test</password>
            <orgId>1</orgId>
            <function>CHANGEPASSWORD</function>
                        <person>
                <userId>admin@yellowfin.com.au</userId>
                <password>test321</password>
            </person>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Reponse Elements

The response returned will contain these parameters:

Response Element	Data Type	Description

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

Reponse Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>f6bdc33348cd76d7898f3bed3e54ff31</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- This example shows a basic request for this call, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- Specify the user using the AdministrationPerson object:

```
AdministrationPerson ap = new AdministrationPerson();

ap.setUserId("john.smith@yellowfin.com.au");
ap.setPassword("test123");
rsr.setPerson(ap);
```

- Once the request is configured, perform the call:

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

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

- The response will contain the StatusCode parameters. See the Response Parameters table above for details.

Complete Example

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

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

```
<%
/*              ws_changepwd.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("CHANGEPASSWORD");

AdministrationPerson ap = new AdministrationPerson();

ap.setUserId("john.smith@yellowfin.com.au");
ap.setPassword("test123");

rsr.setPerson(ap);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

This function resets a user's Yellowfin account and prompts them to change their password. This call will not change the password of the account itself. To change the password, use the CHANGEPASSWORD function.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user Id. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "RESETPASSWD".
Person	AdministrationPerson	Object containing details of the user whose password has to be reset. See table below.

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

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the user whose password is to be reset. This can be the user ID or the email address, depending on the Logon ID method.
Password	String	Password of the above account.

Request Example

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

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.
mi.hof.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <web:remoteAdministrationCall>
         <arg0>
            <loginId>admin@yellowfin.com.au</loginId>
            <password>test321</password>
            <orgId>1</orgId>
            <function>RESETPASSWD</function>
                        <person>
               <userId>admin@yellowfin.com.au</userId>
               <password>test321</password>
            </person>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Reponse Elements

The response returned will contain these parameters:

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

Request Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>beabd94a9ffaa562a8b9ba823fc291a0</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Here is what a basic request for this call looks like, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- Use the AdministrationPerson object to identify the user whose account is to be reset:

```
AdministrationPerson ap = new AdministrationPerson();


ap.setUserId("john.smith@yellowfin.com.au");
rsr.setPerson(ap);
```

- Once the request is configured, perform the call:

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

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

- This call's response will contain the StatusCode parameter. See the Response Parameters table above for more details.

Complete Example

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

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

```
<%
/*              ws_ resetpwd.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("RESETPASSWD");

AdministrationPerson ap = new AdministrationPerson();

ap.setUserId("john.smith@yellowfin.com.au");
rsr.setPerson(ap);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

This call will edit a user's details. It accepts AdministrationPerson as a parameter, which can be used to identify the user, by providing their ID (for example, their email address, or another type of ID depending on the Logon ID method).

The response will contain the AdministrationPerson object with the full details of the user.

Note: This call will not change the password of the user's account. To change the password, use the CHANGEPASSWORD function.

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description

LoginId	String	User ID of the Yellowfin web services administrator who calls this web service. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "UPDATEUSER".
Person	Administration Person	Object containing user details that need to be updated. See table below.

Mandatory parameters to set in the AdministrationPerson object for this function:

AdministrationPerson Element	Data Type	Description
UserId	String	User ID to identify the user whose details are to be changed. This can be the user ID or the email address, depending on the Logon ID method.
You can provide any other parameters related to the user detail that needs to be updated. For example, a new FirstName to change their name, or Status to update their status, etc.		

Request Example

The following SOAP example shows the parameters that you can pass to this call. This example shows the user's status being changed to "Inactive":

```
<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>UPDATEUSER</function>
            <person>
                    <userId>binish.sheikh@yellowfin.com.au</userId>
                    <status>ACTIVE</status>
            </person>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Reponse Elements

The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
Person	AdministrationPerson	Object with the user details updated.

Response Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
     <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
        <return>
           <errorCode>0</errorCode>
           <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
           <messages>Getting user information...</messages>
           <messages>Getting user information...</messages>
           <messages>Web Service Request Complete</messages>
           <person>
              <emailAddress>binish.sheikh@yellowfin.com.au</emailAddress>
              <firstName>Binish</firstName>
              <ipId>13000</ipId>
              <lastName>Sheikh</lastName>
              <roleCode>YFADMIN</roleCode>
              <status>ACTIVE</status>
              <timeZoneCode>AUSTRALIA/BRISBANE</timeZoneCode>
              <userId>binish.sheikh@yellowfin.com.au</userId>
           </person>
           <sessionId>586e172a7c4850bb3edbaaf5264a312d</sessionId>
           <statusCode>SUCCESS</statusCode>
        </return>
     </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

Instructions

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

- Here is what a basic request for this call will look like, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- Then use the AdministrationPerson object to identify the user whose details are to be updated:

```
AdministrationPerson ap = new AdministrationPerson();

ap.setUserId("john.smith@yellowfin.com.au");
```

- Specify what detail is to be changed. For example, we can make a user account inactive:

```
ap.setStatus("INACTIVE");              // This shows that the user "john.smith@yellowfin.com.au" will
not be able to log in

rsr.setPerson(ap);
```

- Once the request is configured, perform the call:

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

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

- This call's response will contain the StatusCode and Person parameters. See the Response Parameters table above for more details.

Complete Example

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

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

```
<%
/*             ws_updateuser.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("UPDATEUSER");


AdministrationPerson ap = new AdministrationPerson();
ap.setUserId("john.smith@yellowfin.com.au");
ap.setStutus("INACTIVE");
rsr.setPerson(ap);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
if ("SUCCESS".equals(rs.getStatusCode()) ) {
        ap = rs.getPerson();
        out.write("User Id: " + ap.getUserId() + "<br>");
        out.write("Password: " + ap.getPassword() + "<br>");
        out.write("First Name: " + ap.getFirstName() + "<br>");
        out.write("Last Name: " + ap.getLastName() + "<br>");
        out.write("Initial: " + ap.getInitial() + "<br>");
        out.write("Salutation Code: " + ap.getSalutationCode() + "<br>");
        out.write("Role Code: " + ap.getRoleCode() + "<br>");
        out.write("Email Address: " + ap.getEmailAddress() + "<br>");
        out.write("Language Code: " + ap.getLanguageCode() + "<br>");
        out.write("IpId: " + ap.getIpId() + "<br>");
        out.write("Time Zone Code: " + ap.getTimeZoneCode() + "<br>");
        out.write("Status: " + ap.getStatus() + "<br>");
} else {
        out.write("Failure");
        out.write(" Code: " + rs.getErrorCode() );
}
%>
```

This web service is used to upload an avatar image for a specified user. The image should be an array of bytes in UTF-8 which you could convert to a String value.

Request Parameters

The following parameters should be passed with this request:

Request Element	Data Type	Description

LoginId	String	An admin account to connect to Yellowfin web services. This can be the user ID or the email address, depending on the Logon ID method. This account must have the “web services” role enabled, and must belong to the default (i.e. primary) org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web service function. Set this to "SETAVATARIMAGE".
Person	AdministrationPerson	This object contains details of the user whose avatar is to be changed. See table below.
Parameters	String[]	An array containing a single element representing the avatar image to be uploaded. The image should be an array of bytes in UTF-8 format, which could be converted to a String.

These are the main parameters that you must set in the AdministrationPerson object for this web service call:

AdministrationPerson Element	Data Type	Description
IpId	Integer	Provide the internal Yellowfin IP ID of the user. This value is stored in the Person parameter's IpId field in Yellowfin's database.

Request Example

Below is a SOAP XML example for this request:

```
<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>SETAVATARIMAGE</function>
            <person>
                        <ipId>5</ipId>
            </person>
            <parameters>
               <string>iVBORw0KGgoAAAANSUhEUgAAAdsAAA ... your image string</string>
            </parameters>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Response Parameters

The returned response will contain the following parameter:

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

Response Example

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

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
         <return>
            <errorCode>0</errorCode>
            <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
            <messages>Web Service Request Complete</messages>
            <sessionId>5fbbf0f2d8a6f4902adde5bfb659fed7</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

- Start with a basic request for this function, which includes logging in as the admin user and specifying the web service call to perform:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- You can specify the user by providing their IP ID:

```
AdministrationPerson ap = new AdministrationPerson();
ap.setIpId(5);

rsr.setPerson(ap);
```

- Select the image file, by using the java.nio.file library:

```
Path path = Paths.get("D:/TMP/fish.jpg"); // existing image file
byte[] data = Files.readAllBytes(path);
byte[] encodeBase64 = java.util.Base64.getEncoder().encode(data);
String img = new String(encodeBase64, "UTF-8");

rsr.setParameters(new String[] {img});
```

- Once the request is configured, perform the call:

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

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

- The response will contain the StatusCode parameter. See the Response Parameters table above for details.

Complete Example

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

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

```
<%
/*                  ws_setavatarimage.jsp                                        */
%>
<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<%@ page import="com.hof.util.*, java.util.*, java.text.*" %>
<%@ page import="com.hof.web.form.*" %>
<%@ page import="com.hof.mi.web.service.*" %>
<%@ page import="java.nio.file.Files" %>
<%@ page import="java.nio.file.Paths" %>
<%@ page import="java.nio.file.Path" %>
<%
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("SETAVATARIMAGE");

AdministrationPerson ap = new AdministrationPerson();
ap.setIpId(5);

rsr.setPerson(ap);

Path path = Paths.get("D:/TMP/fish.jpg"); // existing image file
byte[] data = Files.readAllBytes(path);
byte[] encodeBase64 = java.util.Base64.getEncoder().encode(data);
String img = new String(encodeBase64, "UTF-8");

rsr.setParameters(new String[] {img});

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

User Session Termination

These services can end user's Yellowfin sessions.

This service terminates a Yellowfin session. However, it requires the **LoginSessionId** to be able to log the user out. This is is enough to identify the user, hence the user ID is not required. When a single sign-on is performed with either the LOGINUSER or LOGINUSERNOPASSWORD functions, you can get the LoginSessionId via:

```
String token = response.getLoginSessionId();
```

Save this value, so that you can pass it out to the LOGOUTUSER request later:

```
request.setLoginSessionId(token);
```

Request Elements

The following elements will be passed with this request:

Request Element	Data Type	Description
LoginId	String	Yellowfin web services admin user Id. This can be the user ID or the email address, depending on the Logon ID method. This Yellowfin account must have the “web services” role enabled, and must belong to the Default (i.e. Primary) Org.
Password	String	Password of the above account.
OrgId	Integer	Default (i.e. Primary) organization ID within Yellowfin. Always set this to 1.
Function	String	Web services function. Set this to "LOGOUTUSER".
Person	AdministrationPerson	This object will contain the details of the use who is to be logged out. See the table below.
LoginSessionId	String	Session token of the Yellowfin instance that needs to be terminated. This is enough to identify the user.

Mandatory parameters to set in the AdministrationPerson object for this function:

AdministrationPerson Element	Data Type	Description
UserId	String	User ID to identify the user whose details are to be changed. This can be the user ID or the email address, depending on the Logon ID method.
Password	String	Password of the above account.

Request Example

The following SOAP XML example shows the request for this function using the above parameters:

```
<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>LOGOUTUSER</function>
            <person>
               <userId>admin@yellowfin.com.au</userId>
               <password>test</password>
            </person>
            <loginSessionId>39fb11047affb98c9d081fb48bed0093</loginSessionId>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Reponse Elements

The response returned will contain these parameters:

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

Reponse Example

Here is the resposnse of the above SOAP XML call:

```
 <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>9cad6c76734e329c298e7b15c57a19db</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

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

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- Pass the login session ID:

```
rsr.setLoginSessionId(token);
```

- If the user is logged into multiple Tomcat sessions simultaneously, then you can even specify which session to terminate by setting a parameter, for example:

```
String[] _sessionId = new String[]{sessionId}; // log out by Tomcat session Id (cookies JSESSIONID)
rsr.setParameters(_sessionId);
```

  Only one session should be provided for termination per request. Note that the Tomcat session ID is optional; if omitted, Yellowfin will terminate all of the user's sessions.

- Once the request is configured, perform the call:

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

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

- This call's response will contain the StatusCode parameters. See the Response Parameters table for more details.

Complete Example

Below is a complete example of the LOGOUTUSER function. This script is designed to perform the following steps:

- Call the LOGINUSER service which retrieves the LoginSessionId;
- Configures the login link. You will need to click this link first, to initialize a Yellowfin session for the specifed user. (In our example we will log in the user john.smith@yellowfin.com.au. Make sure that the user you mention, already exists in your Yellowfin instance, or you can even modify the userId.)
- Configure the link to log out. You will need to click this once the session has started.

To use this script for yourself, carry out the following the steps:

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

```
<%
/*              logoutuser.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();

String token = request.getParameter("token");

if (token == null) {

        //login the admin user:

        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("LOGINUSER");

        AdministrationPerson ap = new AdministrationPerson();

        String userId = "john.smith@yellowfin.com.au";

        ap.setUserId(userId);
        ap.setPassword("test");

        rsr.setPerson(ap);

        AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

        if ("SUCCESS".equals(rs.getStatusCode()) ) {
                String token_ = rs.getLoginSessionId();
                out.write("Login by opening the link in a new subtab prior to Logout. The tomcat session must
be initialized...");
                out.write("<BR>Login: <A href='http://localhost:8080/logon.i4?LoginWebserviceId=" + token_ +
"'>");
                out.write("http://localhost:8080/logon.i4?LoginWebserviceId=" + token_ + "</a><br>");
                out.write("<BR>Logout: <A href='http://localhost:8080/test.jsp?token=" + token_ + "&userId=" +
userId + "'>");
                out.write("http://localhost:8080/test.jsp?token=" + token_ + "&userId=" + userId + "</a><br>");
        } else {
                out.write("Failure");
                out.write(" Code: " + rs.getErrorCode() );
                return;
        }
} else {

        //logout the user:

        out.write("Trying to logout " + token + " session...<br>");

        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("LOGOUTUSER");

        rsr.setLoginSessionId(token);

        AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

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

This service is similar to the LOGOUTUSER function in that it logs out a specified user, however this one requires either a user ID (for example, an email address or any other type of ID depending on the Login ID method) or a user's IP ID (that is, the value of the IpId field in the Person table in Yellowfin's database), rather than a login session ID to identify the user.

This call uses the AdministrationPerson object which is used to provide the user ID or IpId.

Request Elements

The following elements will be passed with this request:

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

Set either one of these parameters in the AdministrationPerson object:

AdministrationPerson Element	Data Type	Description
UserId	String	User ID to identify the user to terminate their Yellowfin session. This can be the user ID or the email address, depending on the Logon ID method.
IpId	String	Ip ID of the user whose session is to be terminated.

Request Example

The following XML SOAP example shows a request for this function being passed:

```
<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>LOGOUTBYUSERID</function>
            <person>
               <userId>admin@yellowfin.com.au</userId>
               <password>test</password>
            </person>
         </arg0>
      </web:remoteAdministrationCall>
   </soapenv:Body>
</soapenv:Envelope>
```

Reponse Elements

The response returned will contain these parameters:

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

Request Example

The response of our above SOAP example is shown below:

```
<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>f8a04d7c9530ff18f65f95048e6a4500</sessionId>
            <statusCode>SUCCESS</statusCode>
         </return>
      </ns2:remoteAdministrationCallResponse>
   </S:Body>
</S:Envelope>
```

Instructions

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

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

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- Identify the user to log out, by passing their user ID or IpId:

```
AdministrationPerson ap = new AdministrationPerson();
ap.setUserId(userId);

rsr.setPerson(ap);
```

- If the user is logged into multiple Tomcat sessions simultaneously, then you can specify which session to terminate by setting a parameter, for example:

```
String[] _sessionId = new String[]{sessionId}; // log out by Tomcat session Id (cookies JSESSIONID)
rsr.setParameters(_sessionId);
```

  Only one session should be provided for termination per request. Note that the Tomcat session ID is optional; if omitted, Yellowfin will terminate all of the user's sessions.

- Once the request is configured, perform the call:

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

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

- This call's response will contain the StatusCode parameters. See the Response Parameters table above for more details.

Complete Example

Below is a complete example of this function. This script is designed to perform the following steps:

- Configures the login link. You will need to click this link first, to initialize a Yellowfin session for the specifed user. (In our example we will log in the user john.smith@yellowfin.com.au. Make sure that the user you mention, already exists in your Yellowfin instance, or you can even modify the userId.)
- Configure the link to log out. You will need to click this once the session is set up.

To use this script for yourself, carry out the following the steps:

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

```
<%
/*              test.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();


String userId = request.getParameter("userId");

if (userId == null) {

        //login the admin user:

        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("LOGINUSER");


        AdministrationPerson ap = new AdministrationPerson();

        userId = "john.smith@yellowfin.com.au";

        ap.setUserId(userId);
        ap.setPassword("test");


        rsr.setPerson(ap);

        AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);


        if ("SUCCESS".equals(rs.getStatusCode()) ) {
                String token_ = rs.getLoginSessionId();
                out.write("Login by opening the link in a new subtab prior to Logout. The tomcat session must
be initialized...");
                out.write("<BR>Login: <A href='http://localhost:8080/logon.i4?LoginWebserviceId=" + token_ +
"'>");
                out.write("http://localhost:8080/logon.i4?LoginWebserviceId=" + token_ + "</a><br>");

                out.write("<BR>Logout: <A href='http://localhost:8080/test.jsp?token=" + token_ + "&userId=" +
userId + "'>");
                out.write("http://localhost:8080/test.jsp?token=" + token_ + "&userId=" + userId + "</a><br>");

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

        //logout the user:

        out.write("Trying to logout " + userId + " session...<br>");

        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("LOGOUTBYUSERID");

        AdministrationPerson ap = new AdministrationPerson();
        ap.setUserId(userId);
```

```
        rsr.setPerson(ap);

        AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);


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