Administration Service

Note: all code examples here are **pseudocode**. The actual code you will need to use will differ from environment to environment. The examples here are purely meant as a guide.

Single Sign On

top

This service connects to Yellowfin and retrieves a logon token for a given user. When this token is passed to the Yellowfin Logon URL it will disable the login screen for the authenticated users and their session will being immediately.

```
AdministrationServiceResponse rs = null;
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

AdministrationPerson ap = new AdministrationPerson();
ap.setUserId("admin@yellowfin.com.au");
ap.setPassword("test");
rsr.setPerson(ap);

rs = rssbs.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {

        out.write("Success");
        out.write("URL:  http://<yellowfin-server>/logon.i4?LoginWebserviceId=" + rs.getLoginSessionId());

} else {

        out.write("Failure");

}
```

The LOGINUSER function will Single Sign On a given user into Yellowfin. The contents of the AdministrationPerson object will be used to define the user being logged in.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “LOGINUSER”		Web services function
Person	AdministrationPerson	The AdministrationPerson object holding all of the new user’s details for the user creation process

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 login. This can be the user ID or the email address, depending on the Logon ID method.
Password	String	Password of the new user.

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
LoginSessionId	String	A unique login token. This token is appended to the Login URL to take a user directly into Yellowfin.

Redirecting to Yellowfin with the login token

Using the token received from the webservice call (the contents of AdministrationResponse.LoginSessionId), you can forward the user to the URL:

```
http://<YELLOWFIN-SERVER>/logon.i4?LoginWebserviceId=<TOKEN>
```

This URL will bypass the authentication screen in Yellowfin and take the user directly into Yellowfin.

The token has a limited validity period. It must be used within 5 minutes, and once it has been used, it cannot be used again. To make subsequent calls from a third party application into Yellowfin, you must call the LOGINUSER web service again.

Using the token with the JavaScript API

The SSO token can also be used with embedded JavaScript API widgets. The token is added to the scriptlet URL like this:

```
<script type="text/javascript" src="http://localhost/JsAPI?dashUUID=e9a6ab0a-bcb0-4fe6-9663-
4dd33e58f08e&token=<TOKEN>"></script>
```

Specifying Session Options

You can pass variables/switches that toggle functionality only for the session created via this Single Sign On request. These options can be enabled by passing them via the Parameters attribute in the AdministrationRequest, or by appending them to the redirection URL.

Parameters are added to the AdministrationRequest.Parameters attribute as a list of key value pairs, like:

```
YFTOOLBAR=TRUE
```

```
ENTRY=DASHBOARD
```

Options passed on the login URL look like:

```
http://<yellowfin-server>/logon.i4?LoginWebserviceId=<token>&yftoolbar=true&entry=DASHBOARD
```

Here is a list of the available options:

Parameter Option Key	URL Option Key	Description
YFTOOLBAR	yftoolbar	TRUE/FALSE. Hide the Yellowfin toolbar for this session.

ENTRY	entry	Determine the user's only entry point on completion of the login process. This is used in conjunction with other parameters to determine the content the user is shown. Possible values: • `DASHBOARD` • `REPORTLIST` • `BROWSE` • `BROWSETAB` • `CREATEREPORT` • `EDITREPORT` • `VIEWREPORT` • `ADMINISTRATION` • `EDITDASHBOARD` • `VIEWDASHBOARD` • `VIEWSTORYBOARD` • `TIMELINE`
REPORTID	reportid	Used with `ENTRY = EDITREPORT` or `VIEWREPORT`. This is the internal Id of the report to redirect to.
REPORTUUID	reportuuid	Used with `ENTRY = EDITREPORT` or `VIEWREPORT`. This is the UUID of the rpeort to redirect to.
STORYBOARDUUID	storyboarduuid	Used with `ENTRY = VIEWSTORYBOARD`. This is the UUID of the storyboard to redirect to.
MOBILEDEVICE	mobiledevice	`TRUE/FALSE`. Takes the user to a mobile version of the Yellowfin interface.
REPORTNAME	reportname	Used with `ENTRY = EDITREPORT` or `VIEWREPORT`. This is the name of the report to redirect to.
DISABLESOURCEFILTERS		`TRUE/FALSE`. Disable Source Filter or Client Reference Filters for this user, for this session. This allows the user to see all data. This doesn't work for content based on Client Source Substitution.
DISABLEHEADER	disableheader	`TRUE/FALSE`. Hide the Yellowfin header for this session.
HIDEHEADER	hideheader	Same as `DISABLEHEADER`.
DISABLEFOOTER	disablefooter	`TRUE/FALSE`. Hide the Yellowfin footer for this session.
HIDEFOOTER	hidefooter	Same as `DISABLEFOOTER`.
DISABLESIDENAV	disablesidenav	`TRUE/FALSE`. Hide the Yellowfin side navigation for this session.
HIDESIDENAV	hidesidenav	Same as `DISABLESIDENAV`.
DISABLELOGOFF	disablelogoff	`TRUE/FALSE`. Hide the Yellowfin logoff link for this session.
HIDELOGOFF	hidelogoff	Same as `DISABLELOGOFF`.
REASONCODE	reasoncode	Custom string (80 ASCII characters) that will be dropped into all events generated by this session. Events are dropped in the Event table in the Yellowfin database. This option can also be used with the JavaScript API.
REASONDESCRIPTION	reasondescription	Custom string (2048 ASCII characters) that will be dropped into all events generated by this session. Events are dropped in the Event table in the Yellowfin database. This option can also be used with the JavaScript API.
DASHBOARDID	dashboardid	Used with `ENTRY = EDITDASHBOARD` or `VIEWDASHBOARD`. This is the internal id of the dashboard tab to redirect to.
DASHBOARDUUID	dashboarduuid	Used with `ENTRY = EDITDASHBOARD` or `VIEWDASHBOARD`. This is the internal id of the dashboard tab to redirect to.
FILTER<ID>	filter<ID>	Used with `ENTRY = VIEWREPORT`. Override filter values for the given report. FILTER is appended with the internal ID of the filter, plus the string representation of the filter value. For example: `FILTER2134=MALE` You can pass multiple filters for the one report.

SOURCEFILTER_<FILTERCODE>		Override the source filter values for this user for this session. `SOURCEFILTER_` is appended with the Filter Type Code from the Source Filter setup page, plus the value you wish to give access to. For example: `SOURCEFILTER_COUNTRY=AU` You can pass multiple source filter values for the one session. This option can also be used with the JavaScript API.

CONTENT_<INCLUDE/EXCLUDE>		Override the content access that a user has for this session. `CONTENT_` is appended with either `INCLUDE` or `EXCLUDE`. The value for this key is either the Content Category Code, or the Content Category UUID.
		If you choose to `INCLUDE` content, then these will be the only Content Categories that the user will have access to.
		If you choose to `EXCLUDE` content, then these excluded Content Categories will be removed from the content that the user normally has access to.
		Inclusion and Exclusion of content is based on the Category level. All sub categories with the specified category are included or excluded.
		An example of including/excluding with category codes and UUIDs:
		`CONTENT_INCLUDE=TUTORIAL`
		`CONTENT_INCLUDE=c83357db-8aef-4ec7-ab72-fce34de9ee77`
		`CONTENT_EXCLUDE=TUTORIAL`
		`CONTENT_EXCLUDE=c83357db-8aef-4ec7-ab72-fce34de9ee77`
		It is not logical to use both `INCLUDE` and `EXCLUDE` for the same session.
		This option can also be used with the JavaScript API.

This service is a drop in replacement for `LOGINUSER`, but the password for the user being logged in is not required.

Options are the same as `LOGINUSER` except for:

1. AdministrationRequest.Function will be set to `LOGINUSERNOPASSWORD`
2. AdministrationPerson.Password can be blank

An extra parameter needs to be added to the Configuration table of the Yellowfin database to enable this functionality:

IpOrg	ConfigTypeCode	ConfigCode	ConfigData
1	SYSTEM	SIMPLE_AUTHENTICATION	TRUE

You may need to restart Yellowfin for the database change to take affect.

You will receive a web service arror `26: UNSECURE_LOGIN_NOT_ENABLED` if this configuration option is not added to the Yellowfin database.

User Replication and Managment

top

User replication involves synchronising each user in the OEM 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. Synchronisation is usually performed using web service calls from the OEM application to Yellowfin. This can also be managed manually if users in the OEM 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 OEM application.

Functions

The following code will call the Yellowfin web service to create a user:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

person.setUserId("test@yellowfin.com.au");
person.setPassword("test");
person.setFirstName("Simple");
person.setLastName("Simon");
person.setInitial("S");
person.setSalutationCode("MR");
person.setRoleCode("YFADMIN");
person.setEmailAddress("test@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
rsr.setFunction("ADDUSER");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {

out.write("Success");

} else {

out.write("Failure");
out.write(" Code: " + rs.getErrorCode());

}
```

This code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error message explaining why the user creation process failed.

This function will create a user in Yellowfin. The details in the AdministrationPerson object will be used in the user creation process.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “ADDUSER”		Web services function
Person	AdministrationPerson	The AdministrationPerson object holding all of the new user’s details for the user creation process

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

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
FirstName	String	First name of of the new user
LastName	String	Last name of of the new user
Initial	String	Middle name of the new user

SalutationCode	String	Title of the new user. Possible values include: • DR • MISS • MR • MRS • MS
RoleCode	String	Yellowfin role. The specified role here can be the Org Reference Code (YFADMIN) or the name of the role (Yellowfin Administrator)
EmailAddress	String	Email address of the new user

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service to delete a user:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

person.setUserId("test@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
rsr.setFunction("DELUSER");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {

out.write("Success");

} else {

out.write("Failure");
out.write(" Code: " + rs.getErrorCode());

}
```

This code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error message explaining why the user deletion process failed.

This function will delete a user from Yellowfin. The details in the AdministrationPerson object will be used in the user deletion process.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = "DELUSER" or "DELETEUSER"		Web services function
Person	AdministrationPerson	The AdministrationPerson object holding all of the user's details for the user creation process

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

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

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

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.

The following code will call the Yellowfin web service to retrieve a user's details:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

Person.setUserId("testuser@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
rsr.setFunction("GETUSER");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {

out.write("Success");

} else {

out.write("Failure");
out.write(" Code: " + rs.getErrorCode());

}
```

This code will return an AdministrationPerson object with the user's details and `SUCCESS` in the `rs.getStatusCode()`, otherwise it will return an error message explaining why the process failed.

This function will retrieve the details of a particular user in Yellowfin. The details in the AdministrationPerson object will be used in the retrieval process.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “GETUSER”		Web services function
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process

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

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

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	The AdministrationPerson object holding all of the returned user's details

The following code will call the Yellowfin web service to retrieve a user's details via their internal IpId:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

Person.setIpId(5);

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
rsr.setFunction("GETUSERBYIP");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {

out.write("Success");

} else {

out.write("Failure");
out.write(" Code: " + rs.getErrorCode());

}
```

This code will return an AdministrationPerson object with the user's details and `SUCCESS` in the `rs.getStatusCode()`, otherwise it will return an error message explaining why the process failed.

This function will retrieve the details of a particular user in Yellowfin by looking up their IP ID. The details in the AdministrationPerson object will be used in the retrieval process.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = "GETUSERBYIP"		Web services function
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user's User ID for the retrieval process

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

AdministrationPerson Element	Data Type	Description

IpId|Integer|IP ID of the Yellowfin User|

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	The AdministrationPerson object holding all of the returned user’s details

This function will retrieve users from Yellowfin based on a specific search string. This string is compared against the user’s first name, last name, and email address.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “GETUSERSFROMSEARCH”		Web services function
Parameters	Array (String)	Search string to match against Yellowfin users’ first names, last names, and email address

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	Array (AdministrationPerson)	An array of AdministrationPerson objects. These objects hold the returned users’ details that match the search string

The following code will call the Yellowfin web service to validate a user:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

person.setUserId("admin@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setFunction("VALIDATEUSER");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

This code will return an AdministrationPerson object of the particular user if successful; otherwise it will return an error message explaining why the user validation process failed.

This function will validate a specified Yellowfin user, checking if the user currently exists within the application. The details in the AdministrationPerson object will be used in the user validation process.

Request Element	Data Type	Description

LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “VALIDATEUSER”		Web services function
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process

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

AdministrationPerson Element	**Data Type**	**Description**
UserId	String	User ID of the Yellowfin user. This can be the user ID or the email address, depending on the Logon ID method

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	The AdministrationPerson object holding all of the returned user’s details

The following code will call the Yellowfin web service to validate a user’s password:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

person.setUserId("testuser@yellowfin.com.au");
person.setPassword("test");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
rsr.setFunction("VALIDATEPASSWORD");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

This code will check if the password is expired and will return `FAILURE` in the `rs.getStatusCode()` if it is not, otherwise it will return `SUCCESS`.

This function will validate a Yellowfin user’s password. The details in the AdministrationPerson object will be used in the password validation process.

Request Element	**Data Type**	**Description**
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “VALIDATEUSER”		Web services function
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process

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

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

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

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. For security reasons, the user's password cannot be changed with this web service call, but with a separate CHANGEPASSWORD function (below).

The following code will call the Yellowfin web service to edit a user’s details:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

person.setUserId("testuser");
person.setFirstName("John");
person.setLastName("Doe");
person.setInitial("F");
person.setSalutationCode("MR");
person.setRoleCode("YFADMIN");
person.setEmailAddress("testuser@yellowfin.com.au")

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
rsr.setFunction("UPDATEUSER");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

This code will return an AdministrationPerson object with the user’s details and `SUCCESS` in the `rs.getStatusCode()`, otherwise it will return an error message explaining why the process failed.

This function will update a specified Yellowfin user’s details. The details in the AdministrationPerson object will be used in the update process.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “UPDATEUSER”		Web services function
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process

These are the parameters that you can set in the AdministrationPerson object:

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the Yellowfin user. This can be the user ID or the email address, depending on the Logon ID method
Password	String	Password of the Yellowfin user
FirstName	String	First name of of the Yellowfin user
LastName	String	Last name of of the Yellowfin user
Initial	String	Middle name of the Yellowfin user
SalutationCode	String	Title of the Yellowfin user. Possible values include: • DR • MISS • MR • MRS • MS
RoleCode	String	Yellowfin role. The specified role here can be the Org Reference Code (YFADMIN) or the name of the role (Yellowfin Administrator)
EmailAddress	String	Email address of the Yellowfin user
LanguageCode	String	Two letter code for the preferred language
IpId	Integer	Internal Yellowfin IP ID
TimeZoneCode	String	The TimeZoneCode of the Yellowfin user. See appendix for valid values.
Status	String	Indicates a user status. Can be one of the following: • ACTIVE • INACTIVE • INACTIVEWITHEMAIL

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	The AdministrationPerson object holding all of the updated user’s details

The following code will call the Yellowfin web service and change the password for the specified Yellowfin user:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

person.setUserId("test@yellowfin.com.au");
person.setPassword("testtest");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
rsr.setFunction("CHANGEPASSWORD");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will change a specified Yellowfin user’s password.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “CHANGEPASSWORD”		Web services function
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process

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

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the Yellowfin user. This can be the user ID or the email address, depending on the Logon ID method
Password	String	New password of the Yellowfin user

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

Retrieving Objects Belonging to a User

top

Objects belonging to a user in a Primary or Client Organisation can be retrieved with various web service calls. The objects returned in the response will be dependent on the type of call made in the request.

The following code will call the Yellowfin web service and return all reports with a **Web Services Name** that are accessible for the particular user:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

Person.setUserId("testuser@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if user belongs to a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("GETUSERREPORTS");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

This code will return an Array of AdministrationReport objects in `rs.getReports()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error message explaining why the process failed.

This function will return all reports with a **webservice name** that are accessible for the specified Yellowfin user.

Request Element	Data Type	Description

LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “GETUSERREPORTS”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process

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

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

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
Reports	AdministrationReport	An array of AdministrationReport objects. These objects hold report metadata.

The following code will call the Yellowfin web service and return **all** reports that are accessible for the particular user:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

Person.setUserId("testuser@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if user belongs to a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("GETALLUSERREPORTS");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

This code will return an Array of AdministrationReport objects in `rs.getReports()` and `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error message explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “GETUSERREPORTS”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process

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

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

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
Reports	AdministrationReport	An array of AdministrationReport objects. These objects hold report metadata.

The following code will call the Yellowfin web service and return all **commented** reports that are accessible for the particular user:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

Person.setUserId("testuser@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if user belongs to a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("GETREPORTSWITHCOMMENTS");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

This code will return an Array of AdministrationReport objects in `rs.getReports()` and `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error message explaining why the process failed.

This function will return all commented reports in Yellowfin that are accessible by the specified Yellowfin user.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “GETREPORTSWITHCOMMENTS”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process

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

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

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
Reports	AdministrationReport	An array of AdministrationReport objects. These objects hold report metadata.

The following code will call the Yellowfin web service and return all **report favourites** that are accessible for the particular user:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

Person.setUserId("testuser@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if user belongs to a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("GETFAVOURITES");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

This code will return an Array of AdministrationReport objects in `rs.getReports()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error message explaining why the process failed.

This function will return all the favourite reports of a specified Yellowfin user.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “GETFAVOURITES” or “GETFAVORITES”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process

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

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

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	

Reports	AdministrationReport	An array of AdministrationReport objects. These objects hold report metadata.

The following code will call the Yellowfin web service and return all reports that are in the particular user's **inbox**:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

Person.setUserId("testuser@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if user belongs to a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("GETINBOX");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

This code will return an Array of AdministrationReport objects in `rs.getReports()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error message explaining why the process failed.

This function will return the reports that are in the inbox of a specified Yellowfin user.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = "GETINBOX"	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user's User ID for the retrieval process

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

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

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	
Reports	AdministrationReport	An array of AdministrationReport objects. These objects hold report metadata.

The following code will call the Yellowfin web service and return all dashboard tabs **without reports** that are accessible for that particular user:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

Person.setUserId("testuser@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if user belongs to a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("GETUSERTABS");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

This code will return an Array of AdministrationReportGroup objects in `rs.getReportGroups()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will return a list of dashboard tabs that are accessible by the specified Yellowfin user.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “GETUSERTABS”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process

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

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

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	
ReportsGroups	AdministrationReportGroup	An array of AdministrationReportGroup objects. These objects hold dashboard metadata.

The following code will call the Yellowfin web service and return all dashboard tabs with reports that are accessible for that particular user:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

Person.setUserId("testuser@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if user belongs to a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("GETUSERTABS");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

This code will return an Array of AdministrationReportGroup objects in `rs.getReportGroups()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will return a list of dashboard tabs that are accessible by the specified Yellowfin user, with the reports' metadata loaded as well. The metadata for every report in the dashboard tab is contained within the AdministrationReportGroup object.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = "GETUSERTABSWITHREPORTS"	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user's User ID for the retrieval process

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

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

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	
ReportsGroups	AdministrationReportGroup	An array of AdministrationReportGroup objects. These objects hold dashboard metadata. For this particular function, the reports' metadata is also loaded into this object's GroupReports() parameter.

This function will return a list of reports contained within a specified dashboard tab, that is accessible by a specified user in Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au

Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “LOADTABREPORTS”	String	Web services function
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process
ReportGroup	AdministrationReportGroup	The AdministrationReportGroup object holding the Dashboard Tab ID for the retrieval process

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

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

AdministrationReportGroup Element	Data Type	Description
ReportGroupId	String	Dashboard Tab ID

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	
ReportGroups	AdministrationReportGroup	An array of AdministrationReportGroup objects. These objects hold dashboard metadata. For this particular function, the reports’ metadata is also loaded into this object’s GroupReports() parameter.

Group & Role Administration

top
Groups and Roles can be created and modified with a web service call. The objects returned in the response is dependent on the type of call made in the request.

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

The following code will call the Yellowfin web service and return all available roles within Yellowfin:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;

rsr.setOrgId(new Integer(1));
rsr.setFunction("LISTROLES");

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return an Array of AdministrationRole objects in `rs.getRoles()` and SUCCESS in `rs.getStatusCode()`.

This function will return all the available roles within Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au

Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “LISTROLES”	String	Web services function

The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
Roles	AdministrationRole	An array of AdministrationRole objects. These objects hold Yellowfin Role metadata.

The following code will call the Yellowfin web service and return all available groups within Yellowfin:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if you wish to get groups in a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("LISTGROUPS");

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return an Array of AdministrationGroup objects in `rs.getGroups()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will return all the available user groups within Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “LISTGROUPS”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. The parameter is optional.

The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
Groups	AdministrationGroup	An array of AdministrationGroup objects. These objects hold Yellowfin user group metadata.

The following code will call the Yellowfin web service and return the specified group with its members in Yellowfin:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationGroup group = new AdministrationGroup();

group.setGroupName("Group Name");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if you wish to get the group in a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("GETGROUP");
rsr.setGroup(group);

rs = AdministrationService.remoteAdministrationCall(rsr);
group = rs.getGroup();

AdministrationGroupMember[] groupMembers = group.getGroupMembers();
```

The code will return an AdministrationGroup object in `rs.getGroup()`, an Array of AdministrationGroupMembers in `rs.getGroup().getGroupMembers()`, and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will return a specified Yellowfin group with a list of its members.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = "GETGROUP"	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. The parameter is optional.
Group	AdministrationGroup	The AdministrationGroup object holding the Yellowfin user group's name for the retrieval process

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

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the specified Yellowfin Group

The response returned will contain these parameters:

Response Element	Data Type	Description
StatusCode	String	Status of the web service call. Possible values include: • SUCCESS • FAILURE
Group	Administration Group	An array of AdministrationGroup objects holding group metadata. The group members are also loaded into this object's GroupMembers() parameter.

The following code will call the Yellowfin web service and create the specified group in Yellowfin:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationGroup group = new AdministrationGroup();

group.setGroupName("Group Name");
group.setGroupDescription("Group Description");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if you wish to create the group in a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("CREATEGROUP");
rsr.setGroup(group);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Note: you can also put existing Yellowfin users into this newly created group at the same time by populating an AdministrationGroupMember array of AdministrationPerson objects. Each AdministrationPerson object only needs the `LoginId` variable set, and the AdministrationGroupMember array is saved at `group.setGroupMembers()`.

This function will create a Yellowfin group. Note that you are able to save existing Yellowfin users into the new group in this function as well.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “CREATEGROUP”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. The parameter is optional.
Group	AdministrationGroup	The AdministrationGroup object holding the Yellowfin user group’s name for the retrieval process

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

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the specified Yellowfin group
GroupDescription	String	Business description of the specified Yellowfin group
GroupMembers	Array (AdministrationPerson)	An array of AdministrationPerson objects. This is an optional parameter where you can set to save existing Yellowfin users into this group. Each AdministrationPerson object in this scenario only needs to have its UserId parameter set.

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and include a specified user into a specified group in Yellowfin:

```
AdministrationServiceRequest rsr = new AdminstrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationGroup group = new AdministrationGroup();
AdministrationPerson person = new AdministrationPerson();

group.setGroupName("Group Name");
person.setUserId("test@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if you wish to get the group in a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("INCLUDEUSERINGROUP");
rsr.setGroup(group);
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will include a specified Yellowfin user into a specified group.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “INCLUDEUSERINGROUP”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. The parameter is optional.
Group	AdministrationGroup	The AdministrationGroup object holding the Yellowfin user group’s name for the retrieval process
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User Id for the retrieval process

These are the parameters that you need to set in the AdministrationGroup and AdministrationPerson objects:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the specified Yellowfin group

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

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and exclude a specified user from a specified group in Yellowfin.

Note: this user is not deleted from the group but merely excluded from the group definition. An example of why this would be useful is when:

- John Doe is a member of Group A

- Group A is a member of Group B
- John Doe shouldn’t be a member of Group B, therefore he should be excluded from Group B

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationGroup group = new AdministrationGroup();
AdministrationPerson person = new AdministrationPerson();

group.setGroupName("Group Name");
person.setUserId("test@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if you wish to get the group in a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("EXCLUDEUSERINGROUP");
rsr.setGroup(group);
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will exclude a specified Yellowfin user from a specified group. Note that this user is not deleted from the group but merely excluded from the group definition. An example of why this would be useful is when:

- John Doe is a member of Group A
- Group A is a member of Group B
- John Doe shouldn't be a member of Group B, therefore he is to be excluded from Group B

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “EXCLUDEUSERINGROUP”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. The parameter is optional.
Group	AdministrationGroup	The AdministrationGroup object holding the Yellowfin user group’s name for the retrieval process
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User Id for the retrieval process

These are the parameters that you need to set in the AdministrationGroup and AdministrationPerson objects:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the specified Yellowfin group

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

The response returned will contain these parameters:

Response Element	Data Type	Description

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

The following code will call the Yellowfin web service and delete a specified user from a specified group in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationGroup group = new AdministrationGroup();
AdministrationPerson person = new AdministrationPerson();

group.setGroupName("Group Name");
person.setUserId("test@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if you wish to get the group in a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("DELUSERFROMGROUP");
rsr.setGroup(group);
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will delete a specified Yellowfin user from a specified group. Note that this scenario is different from the one in EXCLUDEUSERINGROUP – in this function a user will be permanently removed from the group.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = "DELUSERFROMGROUP"	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. The parameter is optional.
Group	AdministrationGroup	The AdministrationGroup object holding the Yellowfin user group's name for the retrieval process
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user's User Id for the retrieval process

These are the parameters that you need to set in the AdministrationGroup and AdministrationPerson objects:

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the specified Yellowfin group

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

The response returned will contain these parameters:

Response Element	Data Type	Description

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

The following code will call the Yellowfin web service and modify the specified group in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationGroup group = new AdministrationGroup();

group.setGroupName("Group Name");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if you wish to get the group in a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("MODIFYGROUP");
rsr.setGroup(group);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Note: you can also put existing Yellowfin users into this modified group at the same time by populating an AdministrationGroupMember array of AdministrationPerson objects. Each AdministrationPerson object only needs the `LoginId` variable set, and the AdministrationGroupMember array is saved at `group.setGroupMembers()`.

This function will modify the details for a specified group in Yellowfin. Note that you are able to save existing Yellowfin users into the new group in this function as well.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “MODIFYGROUP”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. The parameter is optional.
Group	AdministrationGroup	The AdministrationGroup object holding the Yellowfin user group’s name for the retrieval process

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

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the specified Yellowfin group
GroupDescription	String	Business description of the specified Yellowfin group
GroupMembers	Array (AdministrationPerson)	An array of AdministrationPerson objects. This is an optional parameter where you can set to save existing Yellowfin users into this group. Each AdministrationPerson object in this scenario only needs to have its LoginId parameter set.

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and delete the specified group in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationGroup group = new AdministrationGroup();

group.setGroupName("Group Name");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
rsr.setOrgId(new Integer(1));
// uncomment line below and set Client Reference ID if you wish to get the group in a client organisation
// rsr.setOrgRef("CLIENTREFERENCEIDHERE");
rsr.setFunction("DELETEGROUP");
rsr.setGroup(group);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will delete a specified group from Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “DELETEGROUP”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. The parameter is optional.
Group	AdministrationGroup	The AdministrationGroup object holding the Yellowfin user group’s name for the retrieval process

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

AdministrationGroup Element	Data Type	Description
GroupName	String	Name of the specified Yellowfin group

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

Client Organisation Functionality

top

Yellowfin contains functionality called Client Organisations, which allows multiple virtual instances of Yellowfin to reside in the same server instance.

Client Organisation functionality can be managed with the available web service calls listed below.

The following code will call the Yellowfin web service and list all client organisations within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("LISTCLIENTS");

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return an Array of AdministrationClientOrg objects and `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will list all client organisations available within Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “LISTCLIENTS”	String	Web services function

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and get the specified client organisation within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationClientOrg aco = new AdministrationClientOrg();

aco.setClientReferenceId("CLIENTREFERENCEIDHERE");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("GETCLIENT");
rsr.setClient(aco);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return an AdministrationClientOrg object in `rs.getClient()` and `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will retrieve details for a specified client organisation within Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “GETCLIENT”	String	Web services function

Client	AdministrationClientOrg	The AdministrationClientOrg object holding the Yellowfin client reference ID for the retrieval process

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

AdministrationClientOrg Element	Data Type	Description
ClientReferenceId	Integer	Client Reference ID for the specified client organisation

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and create the specified client organisation within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationClientOrg aco = new AdministrationClientOrg();

aco.setClientName("CLIENTNAME");
aco.setClientReferenceId("CLIENTREFERENCEIDHERE");
aco.setTimeZoneCode("AUSTRALIA/SYDNEY");
aco.setDefaultOrg(false);

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("CREATECLIENT");
rsr.setClient(aco);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

The function will create the specified client organisation in Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “CREATECLIENT”	String	Web services function
Client	AdministrationClientOrg	The AdministrationClientOrg object holding the Yellowfin client reference ID for the retrieval process

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

AdministrationClientOrg Element	Data Type	Description
ClientName	String	Name of the client organisation
ClientReferenceID	String	Client Reference ID which is the unique ID used to identify a client
TimeZoneCode	String	A client organisation’s local time zone code.
DefaultOrg	Boolean	Boolean value that returns true if this organisation is the primary organisation

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and delete the specified client organisation within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationClientOrg aco = new AdministrationClientOrg();

aco.setClientReferenceId("CLIENTREFERENCEIDHERE");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("DELETECLIENT");
rsr.setClient(aco);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will delete the specified client organisation in Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “DELETECLIENT”	String	Web services function
Client	AdministrationClientOrg	The AdministrationClientOrg object holding the Yellowfin client reference ID for the retrieval process

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

AdministrationClientOrg Element	Data Type	Description
ClientReferenceId	Integer	Client Reference ID for the specified client organisation

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and update the specified client organisation within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationClientOrg aco = new AdministrationClientOrg();

aco.setClientName("CLIENTNAME");
aco.setClientReferenceId("CLIENTREFERENCEIDHERE");
aco.setTimeZoneCode("AUSTRALIA/SYDNEY");
aco.setDefaultOrg(false);

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("UPDATECLIENT");
rsr.setClient(aco);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

The following code will call the Yellowfin webservice and update the specified client organisation within Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “UPDATECLIENT”	String	Web services function
Client	AdministrationClientOrg	The AdministrationClientOrg object holding the Yellowfin client reference ID for the retrieval process

These are the parameters that you can set in the AdministrationClientOrg object:

AdministrationClientOrg Element	Data Type	Description
ClientName	String	Name of the client organisation
ClientReferenceId	Integer	Client Reference ID for the specified client organisation. Note that this is used for the initial retrieval process, and cannot be modified during the usage of this function
TimeZoneCode	String	A client organisation’s local time zone code.
DefaultOrg	Boolean	Boolean value that returns true if this organisation is the primary organisation

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and list all users belonging to the specified client organisation within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationClientOrg aco = new AdministrationClientOrg();
AdministrationPerson[] people = null;

aco.setClientReferenceId("CLIENTREFERENCEIDHERE");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("LISTUSERSATCLIENT");
rsr.setClient(aco);

rs = AdministrationService.remoteAdministrationCall(rsr);
people = rs.getPeople();
```

The code will return an Array of AdministrationPerson objects in `rs.getPeople()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will list all Yellowfin users belonging to the specified client organisation in Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “LISTUSERSATCLIENT”	String	Web services function
Client	AdministrationClientOrg	The AdministrationClientOrg object holding the Yellowfin client reference ID for the retrieval process

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

AdministrationClientOrg Element	Data Type	Description
ClientReferenceId	Integer	Client Reference ID for the specified client organisation

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	Array (AdministrationPerson)	Array of AdministrationPerson objects. Each object holds Yellowfin user metadata.

The following code will call the Yellowfin web service and list all client organisations accessible by a specified user within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationClientOrg[] clients = null;
AdministrationPerson person = new AdministrationPerson();

person.setUserId("test@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("GETUSERACCESS");
rsr.setPerson(person);

rs = AdministrationService.remoteAdministrationCall(rsr);
clients = rs.getClients();
```

The code will return an Array of AdministrationClientOrg objects in `rs.getClients()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices eg. admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary organisation ID within Yellowfin. Always set this to 1.
Function = “GETUSERACCESS”	String	Web services function
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user's User ID for the retrieval process

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

AdministrationPersonElement	Data Type	Description
UserId	String	User ID of the Yellowfin user. This can be the user ID or the email address, depending on the Logon ID method

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
Clients	Array (AdministrationClientOrg)	Array of AdministrationClientOrg objects. Each object holds Yellowfin client organisation metadata

Adding to Default Org

The following code will call the Yellowfin web service and add access to the default organisation for a specified user within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationClientOrg aco = new AdministrationClientOrg();
AdministrationPerson person = new AdministrationPerson();

person.setUserId("test@yellowfin.com.au");
aco.setDefaultOrg(true);

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("ADDUSERACCESS");
rsr.setPerson(person);
rsr.setClient(aco);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will add access to a specified client organisation for a specified user win Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “ADDUSERACCESS”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process
Client	AdministrationClientOrg	The AdministrationClientOrg object holding the Yellowfin client organisation’s Client Reference ID for the retrieval process

These are the parameters that you need to set in the AdministrationPerson and AdministrationClientOrg objects:

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

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

Adding to Client Org

The following code will call the Yellowfin web service and add access to a specified client organisation for a specified user within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationClientOrg aco = new AdministrationClientOrg();
AdministrationPerson person = new AdministrationPerson();

person.setUserId("test@yellowfin.com.au");
aco.setClientReferenceId("CLIENTREFERENCEIDHERE");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("ADDUSERACCESS");
rsr.setPerson(person);
rsr.setClient(aco);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will add access to a specified client organisation for a specified user win Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “ADDUSERACCESS”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process
Client	AdministrationClientOrg	The AdministrationClientOrg object holding the Yellowfin client organisation’s Client Reference ID for the retrieval process

These are the parameters that you need to set in the AdministrationPerson and AdministrationClientOrg objects:

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

AdministrationClientOrg Element	Data Type	Description
ClientReferenceId	Integer	Client Reference ID for the specified client organisation

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

Remove from Default Org

The following code will call the Yellowfin web service and remove access to the default organisation for a specified user within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationClientOrg aco = new AdministrationClientOrg();
AdministrationPerson person = new AdministrationPerson();

person.setUserId("test@yellowfin.com.au");
aco.setDefaultOrg(false);

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("REMOVEUSERACCESS");
rsr.setPerson(person);
rsr.setClient(aco);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will remove access to a specified client organisation for a specified user in Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “REMOVEUSERACCESS”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process
Client	AdministrationClientOrg	The AdministrationClientOrg object holding the Yellowfin client organisation’s Client Reference ID for the retrieval process

These are the parameters that you need to set in the AdministrationPerson and AdministrationClientOrg objects:

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

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

Remove from Client Org

The following code will call the Yellowfin web service and remove access to a specified client organisation for a specified user within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationClientOrg aco = new AdministrationClientOrg();
AdministrationPerson person = new AdministrationPerson();

person.setUserId("test@yellowfin.com.au");
aco.setClientReferenceId("CLIENTREFERENCEIDHERE");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("REMOVEUSERACCESS");
rsr.setPerson(person);
rsr.setClient(aco);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will remove access to a specified client organisation for a specified user in Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “REMOVEUSERACCESS”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process
Client	AdministrationClientOrg	The AdministrationClientOrg object holding the Yellowfin client organisation’s Client Reference ID for the retrieval process

These are the parameters that you need to set in the AdministrationPerson and AdministrationClientOrg objects:

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

AdministrationClientOrg Element	Data Type	Description
ClientReferenceId	Integer	Client Reference ID for the specified client organisation

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

Export & Import Content

top

The following code will call the Yellowfin web service and obtain all exportable content within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
ContentResource[] cr = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("GETCONTENT");

rs = AdministrationService.remoteAdministrationCall(rsr);
cr = rs.getContentResources();
```

The code will return an Array of ContentResource objects in `rs.getContentResources()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “GETCONTENT”	String	Web services function

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
ContentResources	Array (ContentResource)	Array of Content Resource objects. Each ContentResource object holds export metadata for all exportable content in Yellowfin.

The following code will call the Yellowfin web service and obtain dependencies for a specific Yellowfin artifact:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
ContentResource[] cr = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("GETEXPORTDEPENDENCIES");

rs = AdministrationService.remoteAdministrationCall(rsr);
cr = rs.getContentResources();
```

The code will return an Array of ContentResource objects in `rs.getContentResources()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will obtain dependencies for a specific Yellowfin artifact. Note, you can execute the GETCONTENT webservices call to retrieve all exportable content metadata first. This comes in the form of an array of ContentResource objects which can be then used in this function call to retrieve their respective dependencies.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices

OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “GETEXPORTDEPENDENCIES”	String	Web services function
ContentResources	Array (ContentResource)	Array of ContentResource objects. These objects will hold specific Yellowfin artifact metadata

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

ContentResource Element	Data Type
ResourceName	String
ResourceDescription	String
ResourceType	String
ResourceOrgId	Integer
ResourceCode	String

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
ContentResources	Array (ContentResource)	Array of ContentResource objects. These objects will hold dependency metadata

The following code will call the Yellowfin web service and export the specified artifacts within Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

// It is advisable to run a GETCONTENT web service call beforehand to retrieve the necessary ContentResource
objects
// This list can be copied over to the exportList array below
ContentResource[] exportList;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("EXPORTCONTENT");
rsr.setContentResources(exportList);

rs = AdministrationService.remoteAdministrationCall(rsr);
rbo = rs.getBinaryAttachments();
```

The code will return an Array of ReportBinaryObjects objects in `rs.getBinaryAttachments()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will export the specified artifacts within Yellowfin. Note, you can execute the GETCONTENT webservices call to retrieve all exportable content metadata first.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “EXPORTCONTENT”	String	Web services function

ContentResources	Array (ContentResource)	Array of ContentResource objects. These objects will hold specific Yellowfin artifact metadata

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

ContentResource Element	Data Type
ResourceName	String
ResourceDescription	String
ResourceType	String
ResourceOrgId	Integer
ResourceCode	String

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
BinaryAttachments	Array(ReportBinaryObjects)	Array of ReportBinaryObject objects. These objects will hold the XML for exported artifacts.

The following code will call the Yellowfin web service and read a Yellowfin import file, returning the objects to be imported:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
Byte[] data = <XML import file>;
ContentResource[] cr = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("GETIMPORTCONTENT");
rsr.setParameters( new String[] { Base64.encodeBytes(data) } );

rs = AdministrationService.remoteAdministrationCall(rsr);
cr = rs.getContentResources();
```

The code will return an Array of ContentResource objects in `rs.getContentResources()` and SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Refer to `ws_admin_import.jsp` in your Yellowfin web services directory `Yellowfin\development\examples\web services` for a more detailed example of how this function would work.

This function will read a Yellowfin XML import file and return a list of objects that can be imported into the application. This function will be expecting a Base64 encoded XML file as input.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “GETIMPORTCONTENT”	String	Web services function
Parameters	Array (String)	Array of Base64 encoded chunks of the XML import file.

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
ContentResources	Array (ContentResource)	Array of ContentResource objects. These objects will hold specific Yellowfin artifact metadata

The following code will call the Yellowfin web service and validate the Yellowfin import objects. This function usually follows after the GETIMPORTCONTENT web services call:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
Byte[] data = <XML import file>;
ImportIssue[] ii = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("TESTIMPORTCONTENT");
rsr.setParameters( new String[] { Base64.encodeBytes(data) } );

rs = AdministrationService.remoteAdministrationCall(rsr);
ii = rs.getImportIssues();
```

The code will return an Array of ImportIssue objects in `rs.getImportIssues()` if there are issues with the import file and `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Refer to `ws_admin_import.jsp` in your Yellowfin web services directory `Yellowfin\development\examples\web services` for a more detailed example of how this function would work.

This function will validate a list of Yellowfin import objects. This function usually follows after the GETIMPORTCONTENT webservices call, where it reads a Yellowfin XML import file and returns a list of objects that can be imported into the application. In this function, it performs a validation process for every object in that list. Any issues encountered during the validation process is returned as ImportIssue objects in the response.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1
Function = “TESTIMPORTCONTENT”	String	Web services function
Parameters	Array (String)	An array of Base64 encoded chunks of the XML import file
ImportOptions	Array (ImportOption)	An array of ImportOption objects that controls how content is imported.

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
ImportIssues	Array (ImportIssue)	Array of ImportIssue objects. These objects will contain error messages returned from the validation process

The following code will call the Yellowfin web service and import the specified Yellowfin import objects into the application:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
Byte[] data = <XML import file>;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("IMPORTCONTENT");
rsr.setParameters( new String[] { Base64.encodeBytes(data) } );

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Refer to `ws_admin_import.jsp` in your Yellowfin web services directory `Yellowfin\development\examples\web services` for a more detailed example of how this function would work.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1
Function = "IMPORTCONTENT"	String	Web services function
Parameters	Array (String)	An array of Base64 encoded chunks of the XML import file
ImportOptions	Array (ImportOption)	An array of ImportOption objects that controls how content is imported.

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

Delete Content

top

The following code will call the Yellowfin web service and delete a report in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

// This is a Yellowfin Report UUID
String[] parameters ={
"7368e6d4-6167-4a16-ba52-ffa2440a5c8c"
};

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("DELETEREPORT");
// If the report ID is not set, then the code will look for the UUID in parameters
rsr.setReportId(12345);
rsr.setParameters(parameters);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “DELETEREPORT”	String	Web services function
Parameters	Array (String)	The UUID of the Yellowfin Report. Note, you do not need to set this if ReportId is set
ReportId	Integer	The Yellowfin Report ID. Note, you do not need to set this if the UUID is set in the Parameter array

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and delete a view in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

// This example has a Yellowfin View UUID. The parameter value here can be either the UUID
// or the Yellowfin View ID
String[] parameters ={
"7368e6d4-6167-4a16-ba52-ffa2440a5c8c"
};

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("DELETEVIEW");
rsr.setParameters(parameters);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “DELETEVIEW”	String	Web services function
Parameters	Array (String)	Either the Yellowfin View ID or UUID can be set here

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and delete a data source in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

// This is the Yellowfin Data Source ID
String[] parameters ={
"23456"
};

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("DELETESOURCE");
rsr.setParameters(parameters);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return {`SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “DELETESOURCE”	String	Web services function
Parameters	Array (String)	Either the Yellowfin Data Source ID or UUID can be set here

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and delete a dashboard tab in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

// This is the Yellowfin Dashboard Tab UUID
String[] parameters ={
"7368e6d4-6167-4a16-ba52-ffa2440a5c8c"
};

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("DELETETAB");
rsr.setDashboardTabId(11223);
// If the Dashboard Tab ID is not set, then the code will look for the UUID in parameters
rsr.setParameters(parameters);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	**Data Type**	**Description**
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “DELETETAB”	String	Web services function
Parameters	Array (String)	UUID of the Yellowfin Dashboard Tab. Note that this does not need to be set if the DashboardTabId parameter is set
DashboardTabId	Integer	Dashboard Tab ID

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

Cache Management

top

This function will reload the specified Org Reference Codes within Yellowfin.

Request Element	**Data Type**	**Description**
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “RELOADCODES”	String	Web services function
Parameters	Array (String)	The list of Org Reference Codes

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and clear the geometry cache in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

// This is a Yellowfin View ID
String[] parameters ={
"48910"
};

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("GEOMETRYFLUSH");
rsr.setParameters(parameters);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

The function will clear the geometry cache in Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “GEOMETRYFLUSH”	String	Web services function
Parameters	Array (String)	The list of View IDs from which the geometry objects are cached with

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and remove a view’s cache in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

// This is a Yellowfin View ID
String[] parameters ={
"49283"
};

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("REMOVEVIEW");
rsr.setParameters(parameters);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will remove a view's cache in Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “REMOVEVIEW”	String	Web services function
Parameters	Array (String)	The list of View IDs

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and remove a report’s cached definitions in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

// This is a Yellowfin Report ID
String[] parameters ={
"11111"
};

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("FLUSHREPORT");
rsr.setParameters(parameters);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au

Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “FLUSHREPORT”	String	Web services function
Parameters	Array (String)	The list of Report IDs

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and remove a dashboard’s cached definitions in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

// This is a Yellowfin Dashboard Tab ID
String[] parameters ={
"12345"
};

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("FLUSHTAB");
rsr.setParameters(parameters);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “FLUSHTAB”	String	Web services function
Parameters	Array (String)	The list of Tab IDs

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and reload the licence definitions in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("RELOADLICENCE");

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “RELOADLICENCE”	String	Web services function

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and close the specified data source’s connection pool in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

// This is a Yellowfin Data Source ID
String[] parameters ={
"11111"
};

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("CLOSECONNECTIONPOOL");
rsr.setParameters(parameters);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “CLOSECONNECTIONPOOL”	String	Web services function
Parameters	Array (String)	Yellowfin Data Source ID

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and flush the specified filter's filter cache in Yellowfin:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

// This is a Yellowfin Filter ID
String[] parameters ={
"12345"
};

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("FLUSHCACHEDFILTERCACHE");
rsr.setParameters(parameters);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = "FLUSHCACHEDFILTERCACHE"	String	Web services function
Parameters	Array (String)	Filter ID

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

Miscellaneous

top

This function will check if a specified report is in a specified user's favourites list in Yellowfin.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au

Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “ISUSERFAVOURITE”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process
ReportId	Integer	Unique ID for the report

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

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

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and add a specified report to a specified user’s favourites list:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

person.setUserId("test@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("ADDTOFAVOURITES");
rsr.setPerson(person);
rsr.setReportId(12345);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return SUCCESS in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will add a specified report to a specified user’s report favourites list.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “ADDTOFAVOURITES" OR "ADDTOFAVORITES”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	Administration Person	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process

ReportId	Integer	Unique ID for the report

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

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

The response returned will contain these parameters:

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

The following code will call the Yellowfin web service and remove a specified report to a specified user’s favourites list:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;
AdministrationPerson person = new AdministrationPerson();

person.setUserId("test@yellowfin.com.au");

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("REMOVEFAVOURITE");
rsr.setPerson(person);
rsr.setReportId(12345);

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

This function will remove a specified report to a specified user’s report favourites list.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “REMOVETOFAVOURITES" OR "REMOVETOFAVORITES”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	Administration Person	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process
ReportId	Integer	Unique ID for the report

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

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

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

This function will return all avatar images of specified users within Yellowfin:

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “GETAVATARS”	String Web services function	
Parameters	Array (String)	The Unique IDs of Yellowfin users

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
BinaryAttachments	Array (ReportBinaryObject)	Array of ReportBinary objects holding the Base64 decoded avatar images

This function will set the avatar image of the specified user’s profile within Yellowfin:

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “SETAVATARIMAGE”	String	Web services function
OrgRef	String	Client Reference ID if this function is to be applied for a particular client organisation. This parameter is optional.
Person	AdministrationPerson	The AdministrationPerson object holding the Yellowfin user’s User ID for the retrieval process
Parameters	Array (Bytes)	Base64 encoded avatar image

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

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the Yellowfin user. This can be the user ID or the email address, depending on the Logon ID method
IpId	Integer	Unique IP ID of the Yellowfin user

The response returned will contain these parameters:

Response Element	Data Type	Description

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

The following code will call the Yellowfin web service and approve a report in Yellowfin via the expert approval process:

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();
AdministrationServiceResponse rs = null;

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organisation
rsr.setOrgId(new Integer(1));
rsr.setFunction("APPROVEREPORT");
rsr.setReportId(12345)

rs = AdministrationService.remoteAdministrationCall(rsr);
```

The code will return `SUCCESS` in `rs.getStatusCode()`, otherwise it will return an error explaining why the process failed.

Request Element	Data Type	Description
LoginId	String	Login ID of the account used to connect to Yellowfin webservices i.e admin@yellowfin.com.au
Password	String	Password of the account used to connect to Yellowfin webservices
OrgId	Integer	Primary Organisation ID within Yellowfin. Always set this to 1.
Function = “APPROVEREPORT”	String	Web services function
ReportId	Integer	Report ID

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

Object Definitions

top

AdministrationPerson Element	Data Type	Description
UserId	String	User ID of the Yellowfin user. This can be the user ID or the email address, depending on the Logon ID method
Password	String	Password of the Yellowfin user
FirstName	String	First name of of the Yellowfin user
LastName	String	Last name of of the Yellowfin user
Initial	String	Middle initial of the Yellowfin user

SalutationCode	String	Title of the Yellowfin user. Possible values include: - DR - MISS - MR - MRS - MS
RoleCode	String	Yellowfin role. The specified role here can be the Org Reference Code (YFADMIN) or the name of the role (Yellowfin Administrator)
EmailAddress	String	Email address of the Yellowfin user
LanguageCode	String	Two letter code for the preferred language
IpId	Integer	Internal Yellowfin IP ID
TimeZoneCode	String	The TimeZoneCode of the Yellowfin user.

AdministrationReport Element	Data Type
ReportName	String
ReportDescription	String
ReportId	Integer
ExecutionObject	String
ReportCategory	String
ReportSubCategory	String
BirtData	String
SourceName	String
SourceId	Integer
AuthoringMode	String
ReportTemplate	String
DataOutput	String
DashboardEnabled	Boolean
ViewId	Integer
ViewName	String
ViewDescription	String
LastModifierName	String
LastModifierId	Integer
LastModifiedDate	Date
PublishDate	Date
DeliveryMode	String
LastRunTime	Integer
AverageRunTime	Integer
RoleCode	String
ChartTypeCode	String
Usage	Integer

AdministrationReportGroup Element	Data Type	Description
ReportGroupName	String	

ReportGroupDescription	String	
ReportGroupId	Integer	
ReportGroupStatus	String	
ReportGroupType	String	
ReportGroupInternalReference	String	
GroupReports	Array (AdministrationReports)	Array of AdministrationReport objects which hold report metadata. These are the reports that make up the portlets within the dashboard tab
AssociatedReports	Array (Integer)	

AdministrationReportGroup Element	Data Type	Description
RoleName	String	The name of the Yellowfin Role
RoleDescription	String	The business description of the Yellowfin Role
RoleCode	String	The internal code for the Yellowfin Role that is generated and referenced in the database

AdministrationGroup Element	Data Type	Description
GroupName	String	
GroupDescription	String	
GroupId	Integer	
GroupStatus	String	
GroupInternalReference	String	
GroupMembers	Array (AdministrationGroupMember)	An array of AdministrationGroupMember objects. These objects hold group member metadata

AdministrationClientOrg Element	Data Type	Description
ClientName	String	Name of the client organisation
ClientReferenceID	String	Client Reference ID which is the unique ID used to identify a client
TimeZoneCode	String	A client organisation’s local time zone code. See appendix for valid values.
DefaultOrg	Boolean	Boolean value that returns true if this organisation is the primary organisation

ReportBinaryObject Element	Data Type	Description
Key	String	Unique key used for Binary Object storage
ContentType	String	MIME type for the Binary Object
Data	Array (Byte)	Raw data for Binary Object

ContentResource Element	Data Type
ResourceName	String
ResourceDescription	String
ResourceId	Integer
ResourceUUID	Integer
ResourceType	String
ResourceOrgId	Integer

ResourceCode	String

ImportOption Element	Data Type	Description
ItemIndex	Integer	The index of the item that this option is attached too.
OptionKey	String	The type of option.
OptionValue	String	A value associated with this option.

<table>
<tr><th>OptionKey</th><th>OptionValue</th><th>Task</th></tr>
<tr><td>SKIP</td><td>True
False</td><td>Skip an item in the import file.</td></tr>
<tr><td>OPTION</td><td>ADD</td><td>Import an item from the import file.</td></tr>
<tr><td>OPTION</td><td>REPLACE</td><td>Replace an existing item with an item from the import file.
<b>Note:</b> this requires an existing record to be replaced.</td></tr>
<tr><td>EXISTING</td><td>Content UUID</td><td>This requries the internal ID of the content you are replacing. For example, if the item is a View, then this value will be an existing ViewId. If the item is a Category or Sub Category it will be the CategoryCode rather than an ID.</td></tr>
<tr><td>DATABASETYPECODE</td><td>
<table>
<tr><th>Database System</th><th>Database Code</th></tr>
<tr><td>Generic ODBC Source</td><td>GENERICODBC</td></tr>
<tr><td>Microsoft Access</td><td>MSACCESS</td></tr>
<tr><td>DB2 AS400</td><td>AS400</td></tr>
<tr><td>Cache DB</td><td>CACHEDB</td></tr>
<tr><td>Columnar DB</td><td>COLUMNAR</td></tr>
<tr><td>DB2</td><td>DB2</td></tr>
<tr><td>Connection Cloud</td><td>CONNECTIONCLOUD</td></tr>
<tr><td>Derby</td><td>DERBY</td></tr>
<tr><td>DIAPRISM</td><td>DIAPRISM</td></tr>
<tr><td>FileMaker Pro</td><td>FILEMAKER</td></tr>
<tr><td>Firebird</td><td>FIREBIRD</td></tr>
<tr><td>FoxPro</td><td>FOXPROODBC</td></tr>
<tr><td>H2</td><td>H2</td></tr>
<tr><td>Hive</td><td>HIVE</td></tr>
<tr><td>HSQLDB</td><td>HSQLDB</td></tr>
<tr><td>Informix</td><td>INFORMIX</td></tr>
<tr><td>Ingres</td><td>INGRES</td></tr>
<tr><td>Vectorwise</td><td>VECTORWISE</td></tr>
<tr><td>Interbase</td><td>INTERBASE</td></tr>
<tr><td>Lucid</td><td>LUCIDDB</td></tr>
</table>
</td><td>Override the details of a data source type during import.</td></tr>
</table>

MonetDB	MONETDB
MySQL	MYSQL
Netezza	NETEZZA
Lotus Notes	NOTES
NouDB	NOUDB
Progress OpenEdge	OPENEDGE
Oracle	ORACLE
Pervasive	PERVASIVE
PostgresSQL	POSTGRESQL
Progress	PROGRESS
SAP HANA	SAPHANA
SAS	SASJDBC
SqLite	SQLITE
Microsoft SQL Server	MSSQLSERVER
Sybase ASA / Sybase SQL Anywhere	SYBASEASA
Sybase ASE	SYBASEASE
Sybase IQ	SYBASEIQ
TeraData	TERADATA
Universe	UNIVERSE
Vertica	VERTICA

DRIVER	JDBC Driver classname	Override the details of a data source host during import.
URL	A JDBC URL	Override the details of a data source URL during import.
DATASOURCE	XMLA Data Source	Override the details of an XMLA data source during import.
CATALOG	XMLA Catalog	Override the details of an XMLA data source catalog during import.
USERNAME	Source Username	Override the details of a data source username during import.
PASSWORD1	Source Password	Override the details of a data source password during import. **Note:** this requires PASSWORD2 to be set as well.
PASSWORD2	Password Validation	Override the details of a data source password during import.
SOURCE	SOURCE<SourceId>	Attach a view to another source. An example of the value might be SOURCE39003 for a source with Id 39003.
CACHEVIEW	True False	Toggle whether a cached view is still cached on import.
CACHESOURCE	SOURCE<SourceId>	Select a source to cache the view against on import. An example of the value might be SOURCE39003 for a source with Id 39003.
CATEGORY	Category Code	Assign a Sub Category code to a parent Category during import.
SOURCE	SOURCE<SourceId>	Assign a report to a source during import. An example of the value might be SOURCE39003 for a source with Id 39003.

VIEW	VIEW<ViewId>	Assign a report to a view during import. An example of the value might be VIEW39003 for a view with Id 39003.
EXECUTIONOBJECTNAME	Name	Assign a web service exection object name during import. Provide a name that uniquely identifies this report via web services.
CATEGORY	CAT<CategoryCode>	Assign a report to an existing report category during import. An example of the value might be CATTUTORIAL for a category with a code of TUTORIAL.
SUBCATEGORY	CAT<CategoryCode>	Assign a report to an existing report sub category, that's a child of the category specified above, during import. An example of the value might be CATTUTORIAL for a category with a code of TUTORIAL.
CATEGORY	IMP<CategoryCode>	Assign a report to a category within the same export file during import. An example of the value might be IMPTUTORIAL for a category with a code of TUTORIAL.
SUBCATEGORY	IMP<CategoryCode>	Assign a report to an existing report sub category within the same export file, that's a child of the category specified above, during import. An example of the value might be IMPTUTORIAL for a category with a code of TUTORIAL.

top