

# Web Services

- [Overview](#)
- [Yellowfin Web Services API](#)

## Overview

[top](#)

Web Services are used for managing communication between an OEM application and Yellowfin. The Web Services are XML based and independent of the programming language used to develop the OEM application.

When developing against Yellowfin Web Services it is possible to generate functional stubs against the WSDL definitions. These definitions can be found at <http://<yellowfin-server>:<port>/services>, eg. <http://localhost:8080/services>.

The functional stub will allow developers to make standard function calls in their native programming language which will directly communicate with the Web Services provided by Yellowfin. The process of creating function stubs should also generate any objects required by the Web Service.

Some of the objects used by the Web Service examples will include:

- `AdministrationServiceRequest` - an object that defines the type of call being made to the web service
- `AdministrationServiceResponse` - an object returned by the web service
- `AdministrationPerson` - an object that contains user information
- `AdministrationGroup` - an object that contains group information

## Yellowfin Web Services API

[top](#)

Yellowfin ships with a JAR file called `yfws-<date>.jar`. This is located Yellowfin installation directory `Yellowfin\development\lib`.

The Yellowfin Web Service API contains pre-generated stubs. This can be used directly in applications that are developed in Java, or other languages that support Java integration, such as Cold Fusion or Lotus Script. This makes integration slightly simpler as each request is not required to be manually generated, as most of the Web Services are wrapped by a standard Java function.

A web service request is of the form:

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

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

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

See [Administration Service](#) or [Report Service](#) for more information.

[top](#)