

YellowfinwebAdministrationScheduleScheduleFrequencyAdministrationScheduleScheduleFrequencyScheduleFrequencyAdministrationScheduleScheduleFrequencyYellowfinclassclassJava

AdministrationScheduleIsActiveBean

webOrgId

LoginId	String	Yellowfin WebIDIDID Yellowfin Yellowfin Web
Password	String	
OrgId	Integer	YellowfinID1
Function	String	Web LISTSCHEDULES

SOAP XML

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

StatusCode	String	Web - SUCCESS - FAILURE
Schedules	AdministrationSchedule[]	AdministrationSchedule

SOAP

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <schedules xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <schedules>
          <frequency>
            <frequencyCode>MONDAY</frequencyCode>
            <frequencyUnit>1</frequencyUnit>
            <localRunTime>0</localRunTime>
            <localTimezoneCode>AUSTRALIA/SYDNEY</localTimezoneCode>
          </frequency>
          <lastRunDateTimeGMT>2018-02-25T00:00:00+11:00</lastRunDateTimeGMT>
          <lastRunStatus>SUCCESS</lastRunStatus>
          <nextRunDateTimeGMT>2018-03-12T00:00:00+11:00</nextRunDateTimeGMT>
          <scheduleActive>true</scheduleActive>
          <scheduleDescription>Athlete</scheduleDescription>
          <scheduleUUID>75a2f5b5-162b-49b5-b197-53643f7dc0de</scheduleUUID>
        </schedules>
        <schedules xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <schedules>
          <frequency>
            <frequencyCode>SATURDAY</frequencyCode>
            <frequencyUnit>6</frequencyUnit>
            <localRunTime>0</localRunTime>
            <localTimezoneCode>AUSTRALIA/SYDNEY</localTimezoneCode>
          </frequency>
          <nextRunDateTimeGMT>2018-03-10T00:00:00+11:00</nextRunDateTimeGMT>
          <scheduleActive>false</scheduleActive>
          <scheduleDescription>Common Filters</scheduleDescription>
          <scheduleUUID>fa757330-b4a8-4047-9b96-745a48b1d1b7</scheduleUUID>
        </schedules>
        <schedules xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <schedules>
          <frequency>
            <frequencyCode>SATURDAY</frequencyCode>
            <frequencyUnit>6</frequencyUnit>
            <localRunTime>0</localRunTime>
            <localTimezoneCode>AUSTRALIA/SYDNEY</localTimezoneCode>
          </frequency>
          <lastRunDateTimeGMT>2018-02-19T00:00:00+11:00</lastRunDateTimeGMT>
          <lastRunError>com.hof.util.ActionErrorsException: java.lang.NullPointerException</lastRunError>
          <lastRunStatus>FAILURE</lastRunStatus>
          <nextRunDateTimeGMT>2018-03-10T00:00:00+11:00</nextRunDateTimeGMT>
          <scheduleActive>false</scheduleActive>
          <scheduleDescription>Common Filters</scheduleDescription>
          <scheduleUUID>f732c7a4-b81a-4788-8038-6771229596c1</scheduleUUID>
        </schedules>
        <sessionId>ba906c4149a72b2f3c750467a31adf72</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

Java

- web

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

rsr.setLoginId(this.username);
rsr.setPassword(this.password);
// This is the primary organization
rsr.setOrgId(new Integer(1));

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

-

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

-

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

web

- StatusCodeSchedules

web

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

```

<%
/*
 * LISTSCHEDULES Example. ws_listschedules.jsp
 * A more complete example can be found in ws_admin_schedule_management.jsp
 */
%>

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

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

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

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

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    AdministrationSchedule[] schedules = rs.getSchedules();
    out.write("Loaded " + schedules.length + " schedules: <br>");

    for (AdministrationSchedule as: schedules) {
        out.write("Schedule " + as.getScheduleUUID() + "<br>");
    }
} else {
    out.write("Failure");
    out.write(" Code: " + rs.getErrorCode() );
}
%>

```

web

LoginId	String	Yellowfin WebIDIDID Yellowfin Yellowfin Web
Password	String	
OrgId	Integer	YellowfinID1
Function	String	Web LOADCHEDULE
Parameters	String[]	UUIDUUID

StatusCode	String	Web • SUCCESS • FAILURE
Schedules	AdministrationSchedule []	AdministrationSchedule

Java

- web

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- UUID

```
// This is the Yellowfin Schedule UUID. Adjust this value
String[] parameters ={
"SOME_UUID"
};
rsr.setParameters(parameters);
```

-

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

web

- StatusCodeSchedules

web

1. **ws_loadschedule.jsp**
2. rootYellowfin/appserver/webapps/ROOT
3. UUID
4. **http://<host>:<port>/ws_loadschedule.jsp**

```

<%
/*
 * LOADSCHEDULE Example.    ws_loadschedule.jsp.
 * A more complete example can be found in ws_admin_schedule_management.jsp
 */
%>

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

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

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

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

// existing Schedule UUID to load. Adjust this value
String[] parameters = {
    "SOME_UUID"
};
rsr.setParameters(parameters);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    AdministrationSchedule schedule = rs.getSchedule();
    out.write("Loaded schedule: " + schedule.getScheduleUUID() + "<br>");

    out.write("Schedule Type: " + schedule.getScheduleTypeCode() + "<br>");
    out.write("Description: " + schedule.getScheduleDescription() + "<br>");
    out.write("Is Active: " + schedule.isScheduleActive() + "<br>");
    out.write("Last Run Status: " + schedule.getLastRunStatus() + "<br>");
    out.write("Last Run Error: " + schedule.getLastRunError() + "<br>");
    out.write("Last Run Date: " + schedule.getLastRunDateTimeGMT() + "<br>");
    out.write("Next Run Date: " + schedule.getNextRunDateTimeGMT() + "<br>");

    // Some schedule types have extra information that you can access, see reference for details
    if (schedule instanceof ReportRefreshSchedule) {
        ReportRefreshSchedule rrs = (ReportRefreshSchedule)schedule;
        out.write("Report To Refresh: " + rrs.getReportId() + "<br>");
    }

    // these values all have different meanings depending on FrequencyType, see reference for details
    out.write("Frequency Type: " + schedule.getFrequency().getFrequencyTypeCode() + "<br>");
    out.write("Frequency Code: " + schedule.getFrequency().getFrequencyCode() + "<br>");
    out.write("Frequency Unit: " + schedule.getFrequency().getFrequencyUnit() + "<br>");
    out.write("Frequency Local Time: " + schedule.getFrequency().getLocalRunTime() + "<br>");
    out.write("Frequency Local Timezone: " + schedule.getFrequency().getLocalTimezoneCode() + "<br>");
} else {
    out.write("Failure");
    out.write(" Code: " + rs.getErrorCode() );
}
%>

```

LoginId	String	Yellowfin WebIDIDID YellowfinYellowfin Web
Password	String	
OrgId	Integer	YellowfinID1
Function	String	WebSAVESCHEDULE
Schedules	AdministrationSchedule	

AdministrationSchedule

AdministrationSchedule		
ScheduleUUID	String	UUID
Frequency	ScheduleFrequency	ScheduleFrequency

ScheduleFrequency

FrequencyTypeCode	String	
FrequencyCode	String	
FrequencyUnit	Integer	
LocalRunTime	Integer	0
LocalTimezoneCode	String	Java

StatusCode	String	Web - SUCCESS - FAILURE
Schedules	AdministrationSchedule	

Java

- web

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

- FrequencyisActiveUUID

```
// This is the AdministrationSchedule which should be saved
AdministrationSchedule s = new AdministrationSchedule();
s.setScheduleUUID("SOME_KNOWN_EXISTING_UUID");
s.setActive(true);

// define the frequency information
ScheduleFrequency f = new MinutesFrequency();
f.setMinutes(5);
s.setFrequency(f);

// set the schedule in the request
rsr.setSchedule(s);
```

•

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

web

- StatusCodeSchedules

web

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

```
<%
/*
 * SAVESCHEDULE Example.    ws_saveschedule.jsp.
 * A more complete example can be found in ws_admin_schedule_management.jsp
 */
%>

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

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



```

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

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

// normally you would load a schedule first and do some sort of modification
rsr.setFunction("LOADSCHEDULE");

// existing Schedule UUID to load. Adjust this value
String[] parameters = {
    "SOME_UUID"
};
rsr.setParameters(parameters);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);
AdministrationSchedule editingSchedule = null;
if ("SUCCESS".equals(rs.getStatusCode()) ) {
    editingSchedule = rs.getSchedule();
} else {
    out.write("LOADSCHEDULE Failure");
    out.write(" Code: " + rs.getErrorCode() );
    return;
}

editingSchedule.setScheduleActive(true);

WeeklyFrequency newFreq = new WeeklyFrequency();
newFreq.setDayOfWeek(ScheduleFrequency.MONDAY);
newFreq.setLocalRunTime(3 * 60 * 60); // 9am
newFreq.setLocalTimezoneCode("AUSTRALIA/SYDNEY");

editingSchedule.setFrequency(newFreq);
rsr.setSchedule(editingSchedule);
rsr.setParameters(null);
rsr.setFunction("SAVESCHEDULE");
rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    AdministrationSchedule updatedSchedule = rs.getSchedule();
    out.write("Loaded schedule: " + updatedSchedule.getScheduleUUID() + "<br>");

    out.write("Schedule Type: " + updatedSchedule.getScheduleTypeCode() + "<br>");
    out.write("Description: " + updatedSchedule.getScheduleDescription() + "<br>");
    out.write("Is Active: " + updatedSchedule.isScheduleActive() + "<br>");
    out.write("Last Run Status: " + updatedSchedule.getLastRunStatus() + "<br>");
    out.write("Last Run Error: " + updatedSchedule.getLastRunError() + "<br>");
    out.write("Last Run Date: " + updatedSchedule.getLastRunDateTimeGMT() + "<br>");
    out.write("Next Run Date: " + updatedSchedule.getNextRunDateTimeGMT() + "<br>");

    // Some schedule types have extra information that you can access, see reference for details
    if (updatedSchedule instanceof ReportRefreshSchedule) {
        ReportRefreshSchedule rrs = (ReportRefreshSchedule)updatedSchedule;
        out.write("Report To Refresh: " + rrs.getReportId() + "<br>");
    }

    // these values all have different meanings depending on FrequencyType, see reference for details
    out.write("Frequency Type: " + updatedSchedule.getFrequency().getFrequencyTypeCode() + "<br>");
    out.write("Frequency Code: " + updatedSchedule.getFrequency().getFrequencyCode() + "<br>");
    out.write("Frequency Unit: " + updatedSchedule.getFrequency().getFrequencyUnit() + "<br>");
    out.write("Frequency Local Time: " + updatedSchedule.getFrequency().getLocalRunTime() + "<br>");
    out.write("Frequency Local Timezone: " + updatedSchedule.getFrequency().getLocalTimezoneCode() + "<br>");
} else {
    out.write("Failure");
}

```

```

        out.write(" Code: " + rs.getErrorCode() );
    }
    %>

```

web

LoginId	String	Yellowfin WebIDIDID Yellowfin Yellowfin Web
Password	String	
OrgId	Integer	YellowfinID1
Function	String	Web DELETESCHEDULE
Parameters	String[]	UUIDUUID

SOAP XML

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://webservices.web.mi.hof.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:remoteAdministrationCall>
      <arg0>
        <loginId>admin@yellowfin.com.au</loginId>
        <password>test</password>
        <orgId>1</orgId>
        <function>DELETESCHEDULE</function>
        <parameters>
          <string>75a2f5b5-162b-49b5-b197-53643f7dc0de</string>
        </parameters>
      </arg0>
    </web:remoteAdministrationCall>
  </soapenv:Body>
</soapenv:Envelope>

```

StatusCode	String	Web • SUCCESS • FAILURE

SOAP

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:remoteAdministrationCallResponse xmlns:ns2="http://webservices.web.mi.hof.com/">
      <return>
        <errorCode>0</errorCode>
        <messages>Successfully Authenticated User: admin@yellowfin.com.au</messages>
        <messages>Web Service Request Complete</messages>
        <sessionId>7b8e70f20d25079f86cf26f5712d15f9</sessionId>
        <statusCode>SUCCESS</statusCode>
      </return>
    </ns2:remoteAdministrationCallResponse>
  </S:Body>
</S:Envelope>
```

Java

- web

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

-

```
// This is the Yellowfin Schedule UUID
String[] parameters = {
  "SOME_UUID"
};
rsr.setParameters(parameters);
```

-

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

web

- StatusCode

web

1. **ws_deleteschedule.jsp**
2. rootYellowfin/appserver/webapps/ROOT
3. UUID
4. **http://<host>:<port>/ws_deleteschedule.jsp**

```

<%
/*
 * DELETESCHEDULE Example.          ws_deleteschedule.jsp
 * A more complete example can be found in ws_admin_schedule_management.jsp
 */
%>

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

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

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

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

String scheduleUUID = "SOME_UUID";
// existing Schedule UUID to delete
String[] parameters = {
    scheduleUUID
};
rsr.setParameters(parameters);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("Successfully deleted schedule: " + scheduleUUID);
} else {
    out.write("Failure");
    out.write(" Code: " + rs.getErrorCode() );
}
%>

```

web

LoginId	String	Yellowfin WebIDIDID YellowfinYellowfin Web
Password	String	
OrgId	Integer	YellowfinID1
Function	String	WebRUNSCHEDULENOW
Parameters	String[]	UUIDUUID

StatusCode	String	Web • SUCCESS • FAILURE
Schedules	AdministrationSchedule	

Java

- web

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

-

```
// This is the Yellowfin Schedule UUID
String[] parameters ={
"SOME_UUID"
};
rsr.setParameters(parameters);
```

-

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

web

- StatusCodeSchedules

web

1. **ws_runschedulenow.jsp**
2. rootYellowfin/appserver/webapps/ROOT
3. UUID
4. **http://<host>:<port>/ws_runschedulenow.jsp**

```
<%
/*
 * RUNSCHEDULENOW Example.      ws_runschedulenow.jsp
 * A more complete example can be found in ws_admin_schedule_management.jsp
 */
```

```

%>

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

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

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

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

// existing Schedule UUID to submit for running
String scheduleUUID = "SOME_UUID";
String[] parameters = {
    scheduleUUID
};
rsr.setParameters(parameters);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("Successfully submitted schedule " + scheduleUUID + " to be run.");

    // NOTE: some properties like last run status/date, etc will not yet be updated,
    // since this call only submits the schedule for run and does not wait until it is complete.
    AdministrationSchedule updatedSchedule = rs.getSchedule();
    out.write("Loaded schedule: " + updatedSchedule.getScheduleUUID() + "<br>");

    out.write("Schedule Type: " + updatedSchedule.getScheduleTypeCode() + "<br>");
    out.write("Description: " + updatedSchedule.getScheduleDescription() + "<br>");
    out.write("Is Active: " + updatedSchedule.isScheduleActive() + "<br>");
    out.write("Last Run Status: " + updatedSchedule.getLastRunStatus() + "<br>");
    out.write("Last Run Error: " + updatedSchedule.getLastRunError() + "<br>");
    out.write("Last Run Date: " + updatedSchedule.getLastRunDateTimeGMT() + "<br>");
    out.write("Next Run Date: " + updatedSchedule.getNextRunDateTimeGMT() + "<br>");

    // Some schedule types have extra information that you can access, see reference for details
    if (updatedSchedule instanceof ReportRefreshSchedule) {
        ReportRefreshSchedule rrs = (ReportRefreshSchedule)updatedSchedule;
        out.write("Report To Refresh: " + rrs.getReportId() + "<br>");
    }

    // these values all have different meanings depending on FrequencyType, see reference for details
    out.write("Frequency Type: " + updatedSchedule.getFrequency().getFrequencyTypeCode() + "<br>");
    out.write("Frequency Code: " + updatedSchedule.getFrequency().getFrequencyCode() + "<br>");
    out.write("Frequency Unit: " + updatedSchedule.getFrequency().getFrequencyUnit() + "<br>");
    out.write("Frequency Local Time: " + updatedSchedule.getFrequency().getLocalRunTime() + "<br>");
    out.write("Frequency Local Timezone: " + updatedSchedule.getFrequency().getLocalTimezoneCode() + "<br>");

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

```

false

LoginId	String	Yellowfin WebIDIDID Yellowfin Yellowfin Web
Password	String	
OrgId	Integer	YellowfinID1
Function	String	Web PAUSESCHEDULE
Parameters	String[]	UUIDUUID

StatusCode	String	Web • SUCCESS • FAILURE
Schedules	AdministrationSchedule	

Java

- web

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

-

```
// This is the Yellowfin Schedule UUID
String[] parameters ={
    "SOME_UUID"
};
rsr.setParameters(parameters);
```

-

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

web

- StatusCodeSchedules

web

1. **ws_pauseschedule.jsp**
2. rootYellowfin/appserver/webapps/ROOT
3. UUID
4. **http://<host>:<port>/ws_pauseschedule.jsp**


```

<%
/*
 * PAUSESCHEDULE Example.          ws_pauseschedule.jsp
 * A more complete example can be found in ws_admin_schedule_management.jsp
 */
%>

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

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

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

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

// existing Schedule UUID to submit for running
String scheduleUUID = "SOME_UUID";
String[] parameters = {
    scheduleUUID
};
rsr.setParameters(parameters);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
    out.write("Successfully paused schedule " + scheduleUUID);

    AdministrationSchedule updatedSchedule = rs.getSchedule();
    out.write("Loaded schedule: " + updatedSchedule.getScheduleUUID() + "<br>");

    out.write("Schedule Type: " + updatedSchedule.getScheduleTypeCode() + "<br>");
    out.write("Description: " + updatedSchedule.getScheduleDescription() + "<br>");
    out.write("Is Active: " + updatedSchedule.isScheduleActive() + "<br>");
    out.write("Last Run Status: " + updatedSchedule.getLastRunStatus() + "<br>");
    out.write("Last Run Error: " + updatedSchedule.getLastRunError() + "<br>");
    out.write("Last Run Date: " + updatedSchedule.getLastRunDateTimeGMT() + "<br>");
    out.write("Next Run Date: " + updatedSchedule.getNextRunDateTimeGMT() + "<br>");

    // Some schedule types have extra information that you can access, see reference for details
    if (updatedSchedule instanceof ReportRefreshSchedule) {
        ReportRefreshSchedule rrs = (ReportRefreshSchedule)updatedSchedule;
        out.write("Report To Refresh: " + rrs.getReportId() + "<br>");
    }

    // these values all have different meanings depending on FrequencyType, see reference for details
    out.write("Frequency Type: " + updatedSchedule.getFrequency().getFrequencyTypeCode() + "<br>");
    out.write("Frequency Code: " + updatedSchedule.getFrequency().getFrequencyCode() + "<br>");
    out.write("Frequency Unit: " + updatedSchedule.getFrequency().getFrequencyUnit() + "<br>");
    out.write("Frequency Local Time: " + updatedSchedule.getFrequency().getLocalRunTime() + "<br>");
    out.write("Frequency Local Timezone: " + updatedSchedule.getFrequency().getLocalTimezoneCode() + "<br>");

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

```

web
true

LoginId	String	Yellowfin WebIDIDID Yellowfin Yellowfin Web
Password	String	
OrgId	Integer	YellowfinID1
Function	String	Web RESUMESCHEDULE
Parameters	String[]	UUIDUUID

StatusCode	String	Web • SUCCESS • FAILURE
Schedules	AdministrationSchedule	

Java

- web

```
AdministrationServiceRequest rsr = new AdministrationServiceRequest();

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

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

-

```
// This is the Yellowfin Schedule UUID
String[] parameters ={
    "SOME_UUID"
};
rsr.setParameters(parameters);
```

-

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

web

- StatusCodeSchedules

web

1. **ws_resumeschedule.jsp**
2. rootYellowfin/appserver/webapps/ROOT
3. UUID
4. **http://<host>:<port>/ws_resumeschedule.jsp**

```

<%
/*
 * RESUMESCHEDULE Example. ws_resumeschedule.jsp
 * A more complete example can be found in ws_admin_schedule_management.jsp
 */
%>

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

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

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

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

// existing Schedule UUID to submit for running
String scheduleUUID = "SOME_UUID";
String[] parameters = {
scheduleUUID
};
rsr.setParameters(parameters);

AdministrationServiceResponse rs = adminService.remoteAdministrationCall(rsr);

if ("SUCCESS".equals(rs.getStatusCode()) ) {
out.write("Successfully resumed schedule " + scheduleUUID);

AdministrationSchedule updatedSchedule = rs.getSchedule();
out.write("Loaded schedule: " + updatedSchedule.getScheduleUUID() + "<br>");

out.write("Schedule Type: " + updatedSchedule.getScheduleTypeCode() + "<br>");
out.write("Description: " + updatedSchedule.getScheduleDescription() + "<br>");
out.write("Is Active: " + updatedSchedule.isScheduleActive() + "<br>");
out.write("Last Run Status: " + updatedSchedule.getLastRunStatus() + "<br>");
out.write("Last Run Error: " + updatedSchedule.getLastRunError() + "<br>");
out.write("Last Run Date: " + updatedSchedule.getLastRunDateTimeGMT() + "<br>");
out.write("Next Run Date: " + updatedSchedule.getNextRunDateTimeGMT() + "<br>");

// Some schedule types have extra information that you can access, see reference for details
if (updatedSchedule instanceof ReportRefreshSchedule) {
ReportRefreshSchedule rrs = (ReportRefreshSchedule)updatedSchedule;
out.write("Report To Refresh: " + rrs.getReportId() + "<br>");
}

// these values all have different meanings depending on FrequencyType, see reference for details
out.write("Frequency Type: " + updatedSchedule.getFrequency().getFrequencyTypeCode() + "<br>");
out.write("Frequency Code: " + updatedSchedule.getFrequency().getFrequencyCode() + "<br>");
out.write("Frequency Unit: " + updatedSchedule.getFrequency().getFrequencyUnit() + "<br>");
out.write("Frequency Local Time: " + updatedSchedule.getFrequency().getLocalRunTime() + "<br>");
out.write("Frequency Local Timezone: " + updatedSchedule.getFrequency().getLocalTimezoneCode() + "<br>");

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

```

