Contents

Preface

1 Introduction 1
   Document Purpose 1
   Scope of the Document 1
   Target Audience 1
   Accessing the APIs 1
   Glossary 1

2 Activity Management API Overview 5
   Activity Management API Overview 5
Preface

This preface introduces information sources that can help you use the application and this guide.

Using Oracle Applications

To find guides for Oracle Applications, go to the Oracle Help Center.

Documentation Accessibility

For information about Oracle’s commitment to accessibility, visit the Oracle Accessibility Program website. Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

Contacting Oracle

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit My Oracle Support or visit Accessible Oracle Support if you are hearing impaired.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides. Please take one of the following surveys:

- For web-based user guide, Web-based User Guide Survey
- For tutorial feedback, Tutorial Survey
1 Introduction

Document Purpose

The document is to provide understanding of basic Activity Management API goals, its methods and the relevant SOAP transactions.

Scope of the Document

This document primarily describes the Activity Management API that is used by Oracle Field Service Cloud (former ETAdirect) to exchange activity-related information (send requests and accept responses) with external systems.

Target Audience

The document is intended for developers and programmers working with the OFSC Activity Management API in order to integrate OFSC with external systems.

Accessing the APIs

To access the Oracle Field Service Cloud Activity Management API, you may import the WSDL using the following URL: https://api.etadirect.com/soap/activity/v3/?wsdl.

Note: The Activity Management API has been deprecated and will be removed with the 20A (February 2020) Update. It is recommended to use the REST based Core API for integration.

Glossary

The following terms and explanations should be used for the goals of the document:

<table>
<thead>
<tr>
<th>Term</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate Route</td>
<td>Start the work day</td>
</tr>
<tr>
<td>Term</td>
<td>Explanation</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Activity</td>
<td>Entity of the Oracle Field Service Cloud system that represents any time-consuming activity of the resource</td>
</tr>
<tr>
<td>Activity Status</td>
<td>Dynamic value that corresponds to the state of particular activity execution</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface – a particular set of rules and specifications that software programs follow to communicate and interact with each other</td>
</tr>
<tr>
<td>Bucket</td>
<td>Entity appearing on the resource tree which can contain resources of a defined type and be assigned activities</td>
</tr>
</tbody>
</table>
| Company          | 1) Legal entity, using Oracle Field Service Cloud  
2) Entity that represents a Client in Oracle Field Service Cloud; company is created by Oracle during the process of implementation                                                                                  |
<p>| Customer         | End-customer, entity that benefits from the activity                                                                                                                                                        |
| ETA              | Predicted time at which a resource will arrive at an appointment and start an activity, calculated dynamically from current and historical data                                                               |
| Inventory        | Equipment that can be installed or deinstalled during an activity                                                                                                                                          |
| Linked Activities| Two separate activities related so that the completion or start of one is dependent on the completion or start of the other                                                                                  |
| Not-ordered      | Activity with an unspecified order of execution in a route, so that it can be executed at any time during the working day. Not-ordered activities do not have defined ETAs or delivery windows                                             |
| Ordered          | Activity with a defined place in a route, which must be performed at a specified time of day. The order of activities can be changed; ordered activities can be changed to not-ordered activities, vice-versa                      |
| Property         | Field and field value assigned to an entity in Oracle Field Service Cloud (to user, resource, activity or inventory). There are fields and company-defined properties.                                             |
| Required Inventory| Inventory necessary for completion of an activity                                                                                                                                                         |
| Resource         | Element in the resource tree representing a defined company asset                                                                                                                                       |
| Resource External ID | Company-unique key used to identify a specific resource                                                                                                                                                    |
| Resource Tree    | Hierarchy of company resources, showing &quot;parent-child&quot; relationships                                                                                                                                      |
| Route            | List of activities assigned to a resource for a specific date, or a list of non-scheduled activities assigned to a resource. A route may contain zero or more activities.                                          |</p>
<table>
<thead>
<tr>
<th>Term</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Window</td>
<td>Time frame expected by the customer for an activity as scheduled by the company</td>
</tr>
<tr>
<td>SOAP</td>
<td>Lightweight protocol for exchange of information in a decentralized, distributed environment</td>
</tr>
<tr>
<td>SOAP 1.1</td>
<td>See <a href="http://www.w3.org/TR/2000/NOTE-SOAP-20000508/">http://www.w3.org/TR/2000/NOTE-SOAP-20000508/</a></td>
</tr>
<tr>
<td>SOAP Interface</td>
<td>Interface used to receive requests and return responses via SOAP</td>
</tr>
<tr>
<td>SOAP Client Application</td>
<td>Application running at the Client's site and providing interaction with Oracle Field Service Cloud server via SOAP</td>
</tr>
<tr>
<td>Team</td>
<td>Group of several resources where one or more resources (Team members) assist another resource (team holder)</td>
</tr>
<tr>
<td>Teamwork</td>
<td>Feature allowing resources to assist each other in an activity or on an on-going basis</td>
</tr>
<tr>
<td>User</td>
<td>1) Person using Oracle Field Service Cloud</td>
</tr>
<tr>
<td></td>
<td>2) Entity used for authentication and authorization, allowing people or external software to access Oracle Field Service Cloud</td>
</tr>
<tr>
<td>Work Skill</td>
<td>Qualification required to perform an activity</td>
</tr>
</tbody>
</table>
2 Activity Management API Overview

Activity Management API Overview

Oracle Field Service Cloud (OFSC) Activity Management Application Programming Interface can be used throughout the activity lifecycle. The API enables integration of the OFSC activity management functionality in any software regardless of the platform or technology used, providing efficient tool for dynamic management and update of activity-related properties. Basic entities processed with the Activity Management API are activities and activity properties.

OFSC API Versioning

Version 3 includes the entire functionality of Versions 1 and 2 as well as new features developed in OFSC version 15.8 and later.

Versioning is implemented by appending 'v3' for 'Version 3' to the API URL.

Activity Management API URL:

Version 3 /soap/activity/v3/

OFSC Entities Related to Activity Management API

To understand how activities are processed in OFSC it is necessary to review some OFSC entities. The basic entities are:

User is the entity that corresponds to a person, group of people or software, accessing OFSC with a single login to the system and specified permissions.

Resource is the entity that represents a company asset such that it or its child resources can perform work for the benefit of a company.

Route is one calendar day of one resource with a list of scheduled or non-scheduled activities assigned to the resource. Routes can include both ordered and not-ordered activities.

Activity is any time-consuming action performed by a resource.

Required Inventory is inventory necessary for completion of an activity.

Inventory is equipment that can be installed or deinstalled during an activity.

Service request is a message generated as a result of ‘send Service Request’ operation and assigned to a specific entity in OFSC.

In the course of implementation the system is populated with users and resources.

Correlation between users and resources is defined – resources are assigned to users. One user can be assigned several resources and one resource can be assigned several users.

For each of its shifts the resource is assigned a specific route.
Each route is populated with activities – activities are assigned to the route. Each route can be assigned zero or more activities.

Subject to the predefined activity rules some activities can be shared by several routes (this is called teamwork).

Basic entities correlation

Properties are parameters assigned to user, resource, route, activity and required inventory.

**Note:** Users can manage properties of activities only in the routes of the resources assigned to them, and the other way round – resources’ properties, inventory, activities and routes can be managed only by the users to which such resources are assigned. More than one user can be assigned several resources.

Entities Not Directly Managed With Activity Management API

**User Entity**

In the course of OFSC implementation for a company, OFSC is populated with a set of users. For each user security profiles, display profiles and API profiles are defined. While a security profile and display profile must always be defined for each user, API profiles are to be assigned to users which need to use APIs.
A Security Profile defines if access to a certain part of the system, specific interface or function within the interface is permitted. One user may be assigned several profiles and if at least with one of them access is granted, it is available to the user.

A Display Profile defines if each specific element of a security profile-permitted transaction (property of an entity processed) can be accessed by the user and the level of access granted. One user may be assigned one display profile.

An API Profile defines the user’s ability to use the API functionality. To be able to use a certain API, the user has to have an API profile in which the list of entities and their properties available for management via the API can be configured. On the other hand, the access to the specific API is controlled by permissions (Manage ￿ Company Settings ￿ Permissions ￿ SOAP). In order to be able to use a certain SOAP API method, the user must have the permission for such method set to ‘Show’.

As of API Version 3, the user configuration implemented as security profiles, display profiles and API profiles has been combined into User Types. Each user is assigned one user type defining such user’s access to the system and its components, screens and APIs.

Note: The Activity Management API is not designed to create users or process user settings, but having entered a system as a specific user, the person or software can manage only the activities of resources assigned to such user and perform only the operations permitted with the security profiles defined for such user.

Resource Entity

In the course of OFSC implementation, OFSC is populated with a set of resources – entities representing people or equipment used by a company, so that they or their child resources can perform work for the benefit of the company. For each resource entity specific parameters (properties) are defined.

Resource type represents a predefined company-specific set of rules applied when processing a resource.

Note: The Activity Management API is not designed to create resources or process resource settings, but it can be used to retrieve data on all activities assigned to a resource, to assign activities to and execute activities from a resource’s route, therefore, it is necessary that resource settings comply with the transaction logics.

Route Entity

One workday of one technician defines a route. Any references to the ‘date’ of the route mean the date of the workday start (e.g. if the resource works overnight).

Initially the route is formed in accordance with the resource calendar for every active resource with the working time scheduled for the date. The route of a resource can be filled with activities. When a user to which such resource is assigned has logged into the system, the user can manage the route.

Non-scheduled Route

For all active resources there is a route for the date 3000-01-01 that can be assigned activities, for which no specific performance date is defined. All other routes are referred to as 'scheduled'.

Note: Route processing is directly connected to assignment of activities to the route and their processing. For more details on route processing please refer to Section 2.5.8, Activities in the Route, where activity management is described in more details.
Properties and Fields

Property is a variable associated with some OFSC entity. This is much like a field of an object in most programming languages. Every property has a string label by which it is uniquely identified – it can be thought of as a field name and a value.

Activity Management API processes properties of activities in OFSC. Some properties are initially defined in OFSC and others are client-specific and created in the course of implementation, though as soon as there is an activity property in OFSC, all activities have this property, and Activity Management API can read this property, write to it, and make conditional decisions based on the property value. Properties initially defined in OFSC are addressed as ‘fields’.

File Properties

If a file is assigned to an entity, it is called a file property or file. File data is always encoded using base64 encoding.

Property Visibility

The way a property will be processed by the Activity Management API depends on its visibility. A property can be set to hidden and will not be seen in any way by the user. On the other hand, visible properties can be mandatory for the request to be processed correctly or optional.

Optional: the user can see the property and can optionally manage it. The ‘Required’ column contains ‘No’ for such property.

Mandatory:

- the user can see the property and must define it
- if the transaction contains an invalid mandatory property, the request is rejected with a corresponding error
- if request has no mandatory property, the request is rejected with a corresponding error

The ‘Required’ column contains ‘Yes’ for such property.

‘properties’ Structure

The ‘properties’ structure represents an activity property in the form of a simple name-value pair consisting of the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Yes</td>
<td>string</td>
<td>property name, unique for the corresponding property list</td>
</tr>
<tr>
<td>value</td>
<td>Yes</td>
<td>string</td>
<td>property value, can be an empty string. When an empty string is sent, the value of the existing property is deleted</td>
</tr>
</tbody>
</table>
Activities

Activity Entity Overview

The routes are filled with activities – entities that correspond to time-consuming actions. Upon an activity creation, a set of parameters (properties) is defined for it. The accessibility of the properties is defined with user’s settings. Along with company-defined properties required by the company-specific business logics, the properties defined for an activity include:

- IDs: automatically generated numeric ID
- Customer details: address, phone number, name, language etc.
- Activity Type: one of the predefined company-specific Activity Types that define a set of company-specific rules applied to the activity
- Preferred Resources: set of required/preferred/forbidden resources used for automatic routing
- Activity links: set of dependencies defining the correlation between start/end of one activity and start/end of another activity
- Activity timing details: when the activity should be performed
- Required inventory: inventory necessary to complete the activity

Activity Types

An activity type corresponds to a predefined company-specific set of rules applied when processing an activity. The rules cover the resources the activity can be assigned to, details of its processing and interaction with different modules of OFSC (e.g. statistics, notification).

It is not possible to define activity type and resource details with the Activity Management API but it is possible to define the type of activity processed with the application and to assign the activity to a resource, and thus features of the types may influence activity processing.

Note: for correct activity processing its type settings must correspond to the activity management logics, otherwise errors can occur. As the explanation of possible discrepancy requires deeper understanding of other entities, it is provided in the Resource Type Features Affecting Activity Management.

Preferred Resources

Activities can be assigned to a route of a bucket and then allocated to the routes of its child resources. To fit the activity allocation to the company business needs, each activity can be assigned a set of resource preferences. There are three possible preference levels – required, preferred and forbidden. So if any resource is defined as required, automatic routing must assign activities to the route of such resource. If no required resources are defined, but there are some preferred resources, such activities should be assigned to one of the preferred resources. Activity cannot be assigned to the route of a resource which is forbidden for such activity.

Required Inventory

Required inventory is the inventory necessary to complete a certain activity. If any required inventory is defined for an activity, such required inventory is regarded as one of the criteria of activity assignment to resources. The required inventory is checked against the resource’s inventory to see whether the resource’s inventory is sufficient to complete the activity. If the resource has no required inventory in their pool, the activity will not be assigned to such resource.
Serialized and Non-Serialized Inventory

Serialized inventory consists of individual pieces of inventory which are tracked by a serial number. Non-serialized inventory, on the other hand, does not have a serial number. It has quantity where units of measurement, such as feet, pounds, etc. define how volume and consumption are to be measured. This type of inventory is generic and includes items such as faceplates, wires, etc. Such inventory items of one model or type are interchangeable.

Activity Links

Activity links define correlations between start/end of two activities. If a link is created between two activities, it can define the following conditions to be fulfilled in the performance of such activities:

- sequence (the following basic link types are possible: finish-to-start, start-to-start, simultaneous, related)
- minimal and maximal intervals between sequential activities
- whether the linked activities are to be performed by the same resource or different resources

Activity Timing Details

When an activity is created or modified, its timing details can be defined. They can include:

date – specific date when the activity has to be performed – date of the route to which the activity is assigned:

- must be defined in the course of activity creation
- for an existing activity – cannot be changed using the Activity Management API
- if there is no specific date, the activity should be dated 3000-01-01 (assigned to the non-scheduled route)

SLA window – a date range within which the activity has to be performed (started after the start and complete before the end):

- activity can be re-scheduled only within the route with dates that meet the SLA window requirements
- usually is the time agreed with the customer for the activity performance and is particularly useful for non-scheduled activities
- can be defined in the course of activity creation and can be updated before the activity start

service window – a period of time during the day within which the activity has to be started:

- usually is the time agreed with the customer for the activity performance start
- can be defined in the course of activity creation and can be updated before the activity start

If both SLA window and service window are defined for the activity, it should be performed within their overlap period.

time slot – company-specific labeled service windows that can be defined in the system and referred to (e.g. time slot ‘Lunch’ = service window 12 p.m. – 1 p.m.)

Activities in the Route

When activities are assigned to a route, they are placed in accordance with their Position in Route, using statistically calculated or manually defined duration. Their time stamps – start and end times – are statistically calculated, and since the route start and all the way through its performance they are dynamically updated.

Position in Route

When activities are assigned to a route, they are placed within the route according to their position in route – order in the route in respect to all activities of the route.
Position in route:

- must be defined in the course of activity creation
- can be updated before the activity start
- if position in route is defined for an activity it is referred to as 'ordered'
- if position in route is not defined for an activity it is referred to as 'not-ordered'
- not-ordered activity can be performed at any time of the day

Note:

In the non-scheduled route, the order of activities is not significant, thus when creating a non-scheduled activity any value can be set, e.g. 'position_in_route' = 'first'.

Activity Duration and Time stamps

duration – time the activity performance takes:

- may be defined manually (subject to predefined settings)
- otherwise is statistically calculated
- can be updated after the activity start
- is automatically updated when activity end time is updated (when activity is ended by the user)

activity start time – time when the activity is to start or has started:

- is statistically calculated for all ordered activities assigned to a resource for the day
- is dynamically updated as previous activities are being performed
- is updated the moment the activity has been started

activity end time – time when activity is to end or has ended

- is statistically calculated as activity start time + duration
- is dynamically updated as previous activities are being performed and ended
- can be updated after the activity start
- is updated the moment the activity has been ended

Timing details are used to provide the sequence of activity performance and to ensure appropriate time reporting. Ordered activities must be performed in accordance with their position in route; not-ordered activities can be performed at any time.

Activity Statuses

As the resource performs the activity, it changes its status. Using the activity status, OFSC can define the stage of the activity performance and initiate company-specific notifications and reports.

Pending

- when activity is created in the resource’s route it gets the ‘pending’ status
- assigning and re-assigning a 'pending' activity does not change its status
- a pending activity with a defined place in the route and to be performed in the corresponding moment of the working day is ordered
- an activity that can be performed at any moment during the day is not-ordered
- start time and end time of an ordered pending activity are statistically calculated values
Cancelled

- activity that has not been started and will not be performed is of the 'cancelled' status
- only pending activities can be cancelled

Started

- activity that has been started and is being processed is of the 'started' status
- any not-ordered or the first ordered activity in the route can be 'started'
- only one activity can be started within one route at one moment of time
- start time of a started activity is the time when its status was set to 'started', and its end time is a statistically calculated value

Suspended

- when a started activity is suspended its status is changed to 'pending', the activity becomes not-ordered and a new activity with the 'suspended' status is created duplicating the original activity – its start time is the time when its status was set to 'started' and its end time is the time of suspension

Complete

- activity that has been successfully completed is of the 'complete' status
- any started activity can be completed
- start time of such activity is the time when its status was set to 'started' and end time is the time when its status was set to 'complete'

Not done

- activity that has been started but has not been completed is of 'notdone' status
- any started activity can be set 'notdone'
- start time of such activity is the time when its status was set to 'started' and end time is the time when its status was set to 'notdone'

Reopened

- any activity that was set complete, cancelled or not done can be reopened
- a new not-ordered pending activity will be created

Prework: In addition to regular (initial and reopened) activities, there is prework in OFSC, processed with the Activity Management API. Prework is work necessary to perform a specific activity, it is always created in the 'started' status and can be completed or delayed.
Activity and Route Management by Activity Management API

The Activity Management API provides methods to manage routes and activities creating and processing them and changing their details as follows:

Route and Activity Status Processing

Routes can be activated with the Activity Management API method ‘start_route’.

Route can be modified, when required and allowed by the company business processes, as follows:

- new activities (e.g. prescheduled visits to the office or to the warehouse) can be added to the route and time can be allocated for such activities within the working day with the 'create_activity' method at any stage of the route (whether it is activated or not)
- routes can be rearranged, activities order can be changed (e.g. by the technician to better suit the travel path or technical needs of the activity processing) or the details of the activities in the route can be changed with the 'update_activity' method at any stage of the route (whether it is activated or not)

Activity status can be processed with the corresponding methods: 'start_activity', 'suspend_activity', 'complete_activity', 'cancel_activity', 'delay_activity', 'prework_activity', 'reopen_activity'.

Note: Each step in the work order lifecycle must be in chronological order. At any given time, there may be only one started activity. Activities must start and end in their sequential order.
Route can be deactivated with the ‘end_route’ method which can be applied only if there are no pending or started activities left in the route. Route deactivation means that the resource will not be working any longer during that day. By ending the day of a resource, the user automatically notifies the system that no more activities should be scheduled for the resource on that day.

> **Note:** Activities that were not performed must be rescheduled before the user can end the working day of a resource.

Preferred Resource Processing
Resources to be handled as required, preferred or forbidden for the activity can be defined using the ‘set_resource_preferences’ method.

Activity Links Processing
Consequential and simultaneous dependencies between start/end time of several activities can be defined and removed with the ‘link_activities’ and ‘unlink_activities’ methods, respectively.

Retrieving Activity Data
Subject to the user permissions, it is possible to retrieve details of all activities that have a certain property value with the ‘search_activities’ method, for all activities of a specified route with the ‘get_route’ method or for a specific activity with the ‘get_activity’ method. The ‘get_resource_preferences’ and ‘get_activity_links’ methods can be used to retrieve all resource preferences and links of the specified activity, respectively.

Searching Activities by Multiple Criteria (Available Starting from API Version 2)
The Activity Management API allows retrieving activities matching multiple criteria in a single request, ‘get_activities’. The ‘get_activities’ method returns all activities of the specified resource in the specified date range which match the criteria defined in the request and may also include the specified properties of such activities.

Defining Activity Details
As it has been mentioned, a new activity can be created with the ‘create_activity’ method and activity details can be updated with the ‘update_activity’ method. The activity properties can be defined and updated if the corresponding user permissions have been granted. The activity names are labels of activities used in the Activity Management API.

Activity Properties
As it has been mentioned, some of activity and resource type features can affect performance of Activity Management Interface-based applications. In addition to the fields listed in the following table, other activity fields existing in the system and activity properties defined in the specific company can also be processed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Type</th>
<th>Value</th>
<th>Update Allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>aid</td>
<td>int</td>
<td>activity ID</td>
<td>No</td>
</tr>
<tr>
<td>appt_number</td>
<td>appt_number</td>
<td>string</td>
<td>work order number</td>
<td>Yes</td>
</tr>
<tr>
<td>resource_id</td>
<td>-</td>
<td>string</td>
<td>resource external ID</td>
<td>No</td>
</tr>
<tr>
<td>type</td>
<td>atype</td>
<td>enum</td>
<td>activity type</td>
<td>No</td>
</tr>
</tbody>
</table>
## Chapter 2

### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Type</th>
<th>Value</th>
<th>Update Allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>astatus</td>
<td>enum</td>
<td>activity status</td>
<td>No</td>
</tr>
<tr>
<td>worktype</td>
<td>aworktype</td>
<td>enum</td>
<td>activity work type (if ‘aworktype’ is sent, it must contain the activity type ID, if ‘worktype’ is sent – the activity type label)</td>
<td>No</td>
</tr>
<tr>
<td>workzone</td>
<td>aworkzone</td>
<td>enum</td>
<td>activity work zone</td>
<td>No</td>
</tr>
<tr>
<td>duration</td>
<td>length</td>
<td>int</td>
<td>length (duration) of the activity in minutes</td>
<td>Not allowed for work types with the ‘Define duration manually’ feature enabled, otherwise allowed</td>
</tr>
<tr>
<td>time_slot</td>
<td>time_slot</td>
<td>enum</td>
<td>activity time slot (string label) available only if time slots are defined for the company (OFSC Core Manage &gt; Company Settings &gt; Time Slots)</td>
<td>Yes</td>
</tr>
<tr>
<td>service_window_start</td>
<td>service_window</td>
<td>time</td>
<td>customer service window start time in ( , , and )</td>
<td>Yes</td>
</tr>
<tr>
<td>service_window_end</td>
<td>service_window</td>
<td>time</td>
<td>customer service window end time</td>
<td>Yes</td>
</tr>
<tr>
<td>delivery_window_start</td>
<td>delivery_window</td>
<td>time</td>
<td>activity delivery window start in (</td>
<td>Yes</td>
</tr>
<tr>
<td>delivery_window_end</td>
<td>delivery_window</td>
<td>time</td>
<td>activity delivery window end in</td>
<td>Yes</td>
</tr>
<tr>
<td>sla_window_start</td>
<td>sla_window_start</td>
<td>DateTime</td>
<td>activity SLA window start in YYYY-MM-DD</td>
<td>Yes</td>
</tr>
<tr>
<td>sla_window_end</td>
<td>sla_window_end</td>
<td>DateTime</td>
<td></td>
<td></td>
</tr>
<tr>
<td>confirmed_window_start</td>
<td>ctime_delivered_start</td>
<td>DateTime</td>
<td>arrival interval start time communicated to the customer in</td>
<td>Yes</td>
</tr>
<tr>
<td>confirmed_window_end</td>
<td>ctime_delivered_end</td>
<td>DateTime</td>
<td>arrival interval end time communicated to the customer in</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Note: ‘multiday_activity’ and ‘multiday_activity_segment’ are used starting from API Version 3

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>cname</td>
<td>string</td>
<td>customer’s name</td>
</tr>
</tbody>
</table>
## Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Type</th>
<th>Value</th>
<th>Update Allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer_number</td>
<td>customer_number</td>
<td>string</td>
<td>customer's account number</td>
<td>Yes</td>
</tr>
<tr>
<td>phone</td>
<td>cphone</td>
<td>string</td>
<td>The customer's regular (land) phone number. From version 17.2.1, the phone number is saved in OFSC with the ' +' symbol. For example, if you enter the phone number as +1(234)234-23_42, it is saved in OFSC as +12342342342. In versions before 17.2.1, the phone number is saved as 12342342342.</td>
<td>Yes</td>
</tr>
<tr>
<td>email</td>
<td>cemail</td>
<td>string</td>
<td>customer's email address</td>
<td>Yes</td>
</tr>
<tr>
<td>cell</td>
<td>ccell</td>
<td>string</td>
<td>The customer’s cell phone number. From version 17.2.1, the phone number is saved in OFSC with the ' + ' symbol. For example, if you enter the phone number as +1(234)234-23_42, it is saved in OFSC as +12342342342. In versions before 17.2.1, the phone number is saved as 12342342342.</td>
<td>Yes</td>
</tr>
<tr>
<td>address</td>
<td>caddress</td>
<td>string</td>
<td>customer's address</td>
<td>Yes</td>
</tr>
<tr>
<td>city</td>
<td>ccity</td>
<td>string</td>
<td>customer’s city of residence</td>
<td>Yes</td>
</tr>
<tr>
<td>zip</td>
<td>czip</td>
<td>string</td>
<td>customer’s zip/post code</td>
<td>Yes</td>
</tr>
<tr>
<td>Name</td>
<td>Label</td>
<td>Type</td>
<td>Value</td>
<td>Update Allowed</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>state</td>
<td>cstate</td>
<td>string</td>
<td>customer’s state of residence</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note: this field is used by geocoding and, therefore, must contain a valid state name. Other values will not be resolved correctly by the geocoding server.</td>
<td></td>
</tr>
<tr>
<td>language</td>
<td>clanguage</td>
<td>enum</td>
<td>notification language</td>
<td>Yes</td>
</tr>
<tr>
<td>reminder_time</td>
<td>cmessagetime</td>
<td>int</td>
<td>reminder notification time; how many minutes before the activity start time the customer should be notified</td>
<td>Yes</td>
</tr>
<tr>
<td>time_zone</td>
<td>c_zid</td>
<td>enum</td>
<td>customer’s time zone</td>
<td>Yes</td>
</tr>
<tr>
<td>coord_status</td>
<td>acoord_status</td>
<td>string</td>
<td>whether or not activity coordinates were found</td>
<td>No</td>
</tr>
<tr>
<td>coordx</td>
<td>acoord_x</td>
<td>float</td>
<td>latitude of the activity (of the customer’s location)</td>
<td>Yes</td>
</tr>
<tr>
<td>coordy</td>
<td>acoord_y</td>
<td>float</td>
<td>longitude of the activity (of the customer’s location)</td>
<td>Yes</td>
</tr>
<tr>
<td>start_time</td>
<td>ETA</td>
<td>DateTime</td>
<td>ETA time (for and time when the activity was started)</td>
<td>No</td>
</tr>
<tr>
<td>end_time</td>
<td>end_time</td>
<td>DateTime</td>
<td>predicted or actual end time of activity</td>
<td>No</td>
</tr>
<tr>
<td>date</td>
<td>date</td>
<td>date</td>
<td>activity date in YYYY-MM-DD format</td>
<td>No</td>
</tr>
<tr>
<td>team_id</td>
<td>-</td>
<td>string</td>
<td>external ID of the team-holder – the head resource within a team</td>
<td>No</td>
</tr>
<tr>
<td>unordered</td>
<td>-</td>
<td>enum</td>
<td>returned in the response with value = ‘1’ if there is no specific time within the resource’s route when the activity has to be performed (e.g. to perform first, last, after the first etc.)</td>
<td>No</td>
</tr>
<tr>
<td>position_in_route</td>
<td>position_in_route</td>
<td>int</td>
<td>if there is a specific time within the resource’s route when the activity has to be performed,</td>
<td>Yes</td>
</tr>
<tr>
<td>Name</td>
<td>Label</td>
<td>Type</td>
<td>Value</td>
<td>Update Allowed</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------</td>
<td>------</td>
<td>----------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>master_id</td>
<td>-</td>
<td>string</td>
<td>for segmentable activity segments – ID of the segmentable activity</td>
<td>No</td>
</tr>
<tr>
<td>time_to_complete</td>
<td>-</td>
<td>int</td>
<td>for segmentable activities – time remaining until the activity completion</td>
<td>Yes</td>
</tr>
<tr>
<td>multiday_activity_status</td>
<td>-</td>
<td>enum</td>
<td>for segmentable activities – final status of the entire segmentable activity</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Activity Type Features Affecting Activity Management**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Defines if activities of the type...</th>
<th>If the feature is enabled for an activity...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teamwork</td>
<td>... are teamwork activities</td>
<td>... &quot;team_id&quot; field that corresponds to the ID of the team-holder must be defined (is mandatory) if the field is defined for an activity that is not a teamwork – the field is ignored</td>
</tr>
<tr>
<td>Enable segmenting and extended duration</td>
<td>... are segmentable activities</td>
<td>... its duration must be set manually within the range from 5 minutes to 999 hours and the activity can be split into several segments according to the constraints</td>
</tr>
<tr>
<td>Allow creation in buckets</td>
<td>... can be created in buckets</td>
<td>... it can be assigned to a resource that is a bucket if such resource is defined for an activity with the feature disabled, the command will fail</td>
</tr>
<tr>
<td>Support of not-ordered activities</td>
<td>... can be not-ordered</td>
<td>... not-ordered activity of the type be created if an activity with this feature disabled is defined as not-ordered, the command will fail</td>
</tr>
</tbody>
</table>
Teamwork

One of the activity type features defines if activities of the type can be ‘teamwork’. If a resource is assigned an activity that is a teamwork, it means that the resource is assigned to assist another resource or work in a team. Each team consists of a team-holder – the leader of the team, and team-member(s) – the assisting resources. Teamwork in started status means that team-member(s) is located in the same place with the team-holder, and do exactly the same activity.

Segmentable Activity

If the ‘Enable segmenting and extended duration’ feature is enabled for an activity type, such activity type will support segmentation, that is, automatic splitting into segments according to certain constraints. In this case, the activity duration is always to be defined manually.

Resource Type Features Affecting Activity Management
Inventory

Inventory Entity Overview

Inventory is any equipment which is installed or deinstalled during an activity performance. Inventory can be in the resource pool (referred to as ‘resource inventory’) or be already installed at the customer’s premises (referred to as ‘customer inventory’). Along with properties required by the company-specific business logics, the properties defined for an inventory include:

ID: automatically generated numeric ID

Inventory type: type of the inventory as defined in OFSC. The inventory type marks inventory as serialized (identified by its serial number) or non-serialized (having no serial number).

Inventory status: pool to which the inventory currently belongs (customer, resource, install or deinstall).

Serial number: unique number used for serialized inventory to identify the specific inventory piece.

Quantity: number of inventory (only for non-serialized inventory).

Inventory Management by Activity Management API

The Activity Management API provides methods to manage inventories, creating and processing them and changing their details as follows:

Inventory can be created using the Activity Management API methods ‘create_customer_inventory’ and ‘create_resource_inventory’.

To retrieve data on all inventories in the resource pool or customer pool the ‘get_resource_inventory_list’ and ‘get_customer_inventory_list’ methods are used. At the same time, data on a single specified inventory can be retrieved using the ‘get_inventory’ method.

Single inventories can be updated with the ‘update_inventory’ method and deleted by the ‘delete_inventory’ method. Inventories can be moved between pools using Activity Management API methods:

- the ‘install_inventory’ method moves inventory from the ‘resource’ pool to the ‘install’ pool, while the ‘undo_install_inventory’ method performs a reverse operation
- the ‘deinstall_inventory’ method moves inventory from the ‘customer’ pool to the ‘deinstall’ pool, while the ‘undo_deinstall_inventory’ method performs a reverse operation
- the ‘exchange_inventory’ method moves inventory from the ‘resource’ pool to the ‘install’ pool, simultaneously moving another inventory from the ‘customer’ pool to the ‘deinstall’ pool

Inventory Properties

Inventory-related methods deal with properties of inventory. Inventory properties node contains all company-defined inventory properties visible for the Activity Management API user and the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Type</th>
<th>Value</th>
<th>Update Allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>invid</td>
<td>int</td>
<td>inventory ID</td>
<td>No</td>
</tr>
<tr>
<td>status</td>
<td>invpool</td>
<td>string</td>
<td>inventory pool</td>
<td>Yes</td>
</tr>
</tbody>
</table>
# Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Type</th>
<th>Value</th>
<th>Update Allowed</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>invtype</td>
<td>string</td>
<td>inventory type. When specified, the inventory is marked as serialized or non-serialized according to the type. If “type” is not sent, the inventory is marked as serialized. empty by default</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>serial_number</td>
<td>invsn</td>
<td>string</td>
<td>serial number of serialized inventory. Relevant for inventory types defining serialized inventory</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>change</td>
<td>inv_change_invid</td>
<td>int</td>
<td>ID of the inventory exchange</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>resource_id</td>
<td>inv_pid</td>
<td>string</td>
<td>external ID of the resource processing the inventory</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>activity_id</td>
<td>inv_aid</td>
<td>int</td>
<td>ID of the activity to which the inventory pool belongs (empty for “resource pool”)</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>quantity</td>
<td>Quantity</td>
<td>int</td>
<td>present and relevant only if ‘inventory type’ corresponds to non-serialized inventory</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

## Service Request

**Service Request Entity Overview**

Service request is a message generated as the result of the ‘send Service Request’ operation and assigned to a specific entity in OFSC. Along with properties required by the company-specific business logics, the properties defined for a service request include:

- **ID:** automatically generated numeric ID
- **Type:** type of the request referring to the entity it is assigned to (resource request, inventory request or customer request)
- **Request creation data:** ID of the user which created the request, request creation date and time
- **Entity:** ID of the entity to which the service request is assigned
• Subject: subject of the request
• Body: body of the request

Service Request Management By Activity Management API
The Activity Management API provides methods to manage service requests, creating and processing them as follows:

Service requests can be created using the Activity Management API methods 'create_customer_request', 'create_resource_request' and 'create_inventory_request'.

To retrieve data on all service requests assigned to a specific activity, resource or inventory the 'get_customer_request_list', 'get_resource_request_list' and 'get_inventory_request_list' methods are used. At the same time, data on a single specified inventory can be retrieved using the 'get_request' method.

Service Request Properties
Service Request-related methods deal with properties of a request. Service request properties node contains all company-defined service request properties visible for the Activity Management API user and the following fields:

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Type</th>
<th>Value</th>
<th>Update Allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>srid</td>
<td>string</td>
<td>request ID</td>
<td>No</td>
</tr>
<tr>
<td>type</td>
<td>srtype</td>
<td>string</td>
<td>Request type. Mandatory for 'create' request methods and not used for the rest</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'type' is a configurable property, which defines some rules for service request message scenario and content. Within the document the following three types are used:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PM – resource service request</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SR – customer service request</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EQ – inventory service request</td>
<td></td>
</tr>
<tr>
<td>created</td>
<td>srcreated</td>
<td>DateTime</td>
<td>date and time of request creation in YYYY-MM-DD HH:MM format</td>
<td>No</td>
</tr>
<tr>
<td>user_id</td>
<td>sr_uid</td>
<td>int</td>
<td>ID of the user who created the request</td>
<td>No</td>
</tr>
</tbody>
</table>
### Oracle Field Service Cloud

#### Integrating with Activity Management API

## Chapter 2

### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Type</th>
<th>Value</th>
<th>Update Allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>sr_pid</td>
<td>string</td>
<td>external ID of the resource for whom the request was created</td>
<td>No</td>
</tr>
<tr>
<td>activity_id</td>
<td>sr_aid</td>
<td>int</td>
<td>ID of the activity for which the request was created</td>
<td>No</td>
</tr>
<tr>
<td>inventory_id</td>
<td>sr_invid</td>
<td>int</td>
<td>ID of the inventory for which the request was created</td>
<td>No</td>
</tr>
<tr>
<td>date</td>
<td>srdate</td>
<td>date</td>
<td>date and time of request submission</td>
<td>No</td>
</tr>
<tr>
<td>subject</td>
<td>sr_subject</td>
<td>string</td>
<td>subject of the request empty by default</td>
<td>No</td>
</tr>
<tr>
<td>body</td>
<td>sr_body</td>
<td>string</td>
<td>request message text empty by default</td>
<td>No</td>
</tr>
</tbody>
</table>

### Activity Management API Methods

The Activity Management API provides a set of methods to handle the routes, activities, inventory, service requests and file properties in the system.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Method</th>
<th>Can be used to...</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>route</td>
<td>get_route</td>
<td>… retrieve properties of the activities in the specified route for a specified day</td>
<td>Returns the list of activities in the route and their properties visible to the user</td>
</tr>
<tr>
<td></td>
<td>get_route start_route</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>end_route</td>
<td></td>
<td></td>
</tr>
<tr>
<td>activity</td>
<td>create_activity</td>
<td>… populate the specified route with activities</td>
<td>Creates activities with the specified properties assigned to the specified route</td>
</tr>
<tr>
<td></td>
<td>cancel_activity</td>
<td>cancel_activity … cancel a pending activity before it has been started or cancel a started activity to change it to notdone</td>
<td>Changes the pending activity status to ‘cancelled’ or the started activity status to ‘notdone’ and updates all activity properties correctly specified in the request and permitted to the user</td>
</tr>
<tr>
<td>Entity</td>
<td>Method</td>
<td>Can be used to...</td>
<td>Details</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>prework_activity</td>
<td>prework_activity</td>
<td>... create a prework for activity</td>
<td>Creates prework in OFSC with the ‘started’ status and sets the prework properties correctly specified in the request and permitted to the user</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(applicable for activities with ‘pending’ status in a started route)</td>
<td></td>
</tr>
<tr>
<td>start_activity</td>
<td>start_activity</td>
<td>... set activity status ‘started’ (applicable for activities with ‘pending’ status in a started route)</td>
<td>Changes activity status to ‘started’ and updates all activity properties correctly specified in the request and permitted to the user</td>
</tr>
<tr>
<td>suspend_activity</td>
<td>suspend_activity</td>
<td>... change a started activity status to ‘pending’, make it not-ordered and simultaneously create a suspended activity duplicating the started activity</td>
<td>Changes the activity status to ‘pending’, makes it not-ordered and creates a suspended activity duplicating the properties of the original activity and in that suspended activity updates the properties correctly specified in the request and permitted to the user</td>
</tr>
<tr>
<td>delay_activity</td>
<td>delay_activity</td>
<td>... prolong the predicted activity duration time (applicable for activities and preworks with the ‘started’ status)</td>
<td>Changes activity end time and updates all activity properties correctly specified in the request and permitted to the user</td>
</tr>
<tr>
<td>complete_activity</td>
<td>complete_activity</td>
<td>... set activity status to ‘complete’ (applicable for activities and preworks with ‘started’ status)</td>
<td>Changes activity status to ‘complete’ and updates all activity properties correctly specified in the request and permitted to the user</td>
</tr>
<tr>
<td>reopen_activity</td>
<td>reopen_activity</td>
<td>... reopen canceled, not done or complete activity</td>
<td>Creates a pending activity duplicating the properties of the canceled, not done or complete activity and in those pending activities updates the properties correctly specified in the request and permitted to the user</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(applicable for activity with status cancelled, not done or complete not applicable for prework)</td>
<td></td>
</tr>
<tr>
<td>update_activity</td>
<td>update_activity</td>
<td>... alter activity details (applicable for activities regardless of the status)</td>
<td>Updates all properties correctly specified in the request and permitted to the user</td>
</tr>
<tr>
<td>get_activity</td>
<td>get_activity</td>
<td>... retrieve specified activity details (applicable for activities regardless of the status)</td>
<td>Returns all properties of the specified activity available in the system and permitted to the user</td>
</tr>
<tr>
<td>search_activities</td>
<td>search_activities</td>
<td>... retrieve all activities matching the specified criterion</td>
<td>Returns the list of activities with the specified value in the specified field for the specified time period</td>
</tr>
<tr>
<td>get_activities</td>
<td>get_activities</td>
<td>... search activities by multiple criteria</td>
<td>Returns activities of the specified resource matching the defined search criteria.</td>
</tr>
<tr>
<td></td>
<td>(available starting from API Version 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>get_activity_work_skills</td>
<td>get_activity_work_skills</td>
<td>... retrieve details of specified activity work skills</td>
<td>Returns the list of work skills and their required and preferable qualification level as well as the list of capacity categories if any</td>
</tr>
<tr>
<td>get_multiday_activities</td>
<td>get_multiday_activities</td>
<td>... retrieve the list of segmentable activities assigned to the specified resource</td>
<td>Returns the data of all segmentable activities assigned to the specified resource and, optionally, their properties</td>
</tr>
<tr>
<td>Entity</td>
<td>Method</td>
<td>Can be used to...</td>
<td>Details</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>activity/</td>
<td>get_multiday_activity_parts</td>
<td>retrieve the list of all segments of the specified segmentable activity and the</td>
<td>Returns the data of the specified segmentable activity and the list of all segments into which the segmentable activity is split and, optionally, their properties</td>
</tr>
<tr>
<td>preferences</td>
<td>(available starting from API Version 3)</td>
<td>data of the segmentable activity progress</td>
<td></td>
</tr>
<tr>
<td>activity</td>
<td>set_resource_preferences</td>
<td>define resource preferences</td>
<td>Defines required, preferred and forbidden resources to perform the specific activity</td>
</tr>
<tr>
<td>links</td>
<td>get_resource_preferences</td>
<td>retrieve all resource preferences of the specified activity</td>
<td>Returns all preferences defined for the specified activity</td>
</tr>
<tr>
<td>inventory</td>
<td>link_activities</td>
<td>create links</td>
<td>Defines the specified dependency between two specified activities</td>
</tr>
<tr>
<td></td>
<td>unlink_activities</td>
<td>delete links</td>
<td>Removes the specified dependency between two specified activities</td>
</tr>
<tr>
<td></td>
<td>get_activity_links</td>
<td>retrieve details of all links of specified activity</td>
<td>Returns details of all links defined for the activity specified</td>
</tr>
<tr>
<td></td>
<td>create_customer_inventory</td>
<td>create inventory assigned to the activity</td>
<td>Creates inventory with the properties in the customer pool</td>
</tr>
<tr>
<td></td>
<td>create_resource_inventory</td>
<td>create inventory assigned to the resource</td>
<td>Creates inventory with the properties in the resource pool</td>
</tr>
<tr>
<td></td>
<td>delete_inventory</td>
<td>delete specified inventory</td>
<td>Deletes inventory with the specified ID</td>
</tr>
<tr>
<td></td>
<td>get_resource_inventory_list</td>
<td>retrieve details of all inventory of specified resource</td>
<td>Returns all properties available in the system and permitted to the user of all inventory defined for the specified resource</td>
</tr>
<tr>
<td></td>
<td>get_customer_inventory_list</td>
<td>retrieve details of all inventory of specified activity</td>
<td>Returns all properties available in the system and permitted to the user of all inventory defined for the specified activity</td>
</tr>
<tr>
<td></td>
<td>get_inventory</td>
<td>retrieve data on the specified inventory</td>
<td>Returns all properties available in the system and permitted to the user for the specified inventory</td>
</tr>
<tr>
<td></td>
<td>install_inventory</td>
<td>move the specified inventory to the 'install' pool</td>
<td>Moves the specified inventory from 'resource' pool to 'install' pool and updates the specified properties of the inventory</td>
</tr>
<tr>
<td></td>
<td>undo_install_inventory</td>
<td>reverse the 'install_inventory' operation</td>
<td>Moves the specified inventory from the 'install' pool to the 'resource' pool</td>
</tr>
<tr>
<td></td>
<td>deinstall_inventory</td>
<td>move the specified inventory to the 'deinstall' pool</td>
<td>Moves the specified inventory from 'customer' pool to 'deinstall' pool and updates the specified properties of the inventory</td>
</tr>
<tr>
<td></td>
<td>undo_deinstall_inventory</td>
<td>reverse the 'deinstall_inventory' operation</td>
<td>Moves the specified inventory from the 'deinstall' pool to the 'customer' pool</td>
</tr>
</tbody>
</table>
## Activity Management API Overview

<table>
<thead>
<tr>
<th>Entity</th>
<th>Method</th>
<th>Can be used to...</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>exchange_inventory</td>
<td>... install resource inventory and deinstall customer inventory simultaneously</td>
<td>Moves resource inventory to the 'install' pool and simultaneously moves customer inventory to the 'deinstall' pool. Updates the specified properties of the inventories</td>
<td></td>
</tr>
<tr>
<td>update_inventory</td>
<td>... change the specified inventory properties</td>
<td>Changes values of the specified inventory properties</td>
<td></td>
</tr>
<tr>
<td>file</td>
<td>set_file</td>
<td>... define a file property</td>
<td>Creates a file property with the defined parameters for the defined entity</td>
</tr>
<tr>
<td>get_file</td>
<td>... retrieve details of the specified file property</td>
<td>Returns all available details of the file property</td>
<td></td>
</tr>
<tr>
<td>delete_file</td>
<td>... delete the specified file property</td>
<td>Deletes the file property with the specified label</td>
<td></td>
</tr>
<tr>
<td>required inventory</td>
<td>set_required_inventories</td>
<td>... define required inventory for the activity</td>
<td>Adds required inventory to the activity and sets the properties of such inventory</td>
</tr>
<tr>
<td>get_required_inventories</td>
<td>... retrieve required inventory of the activity</td>
<td>Returns the required inventory set for the specified activity together with its properties</td>
<td></td>
</tr>
<tr>
<td>service request</td>
<td>create_resource_request</td>
<td>... create a service request assigned to a resource in OFSC</td>
<td>Creates a service request with the specified properties assigned to the specified resource</td>
</tr>
<tr>
<td>get_resource_request_list</td>
<td>... retrieve data on all specified service requests assigned to the specified resource in OFSC</td>
<td>Retrieves all service requests assigned to the specified resource together with their properties</td>
<td></td>
</tr>
<tr>
<td>create_customer_request</td>
<td>... create a service request assigned to an activity in OFSC</td>
<td>Creates a service request with the specified properties assigned to the specified activity</td>
<td></td>
</tr>
<tr>
<td>get_customer_request_list</td>
<td>... retrieve data on all specified service requests assigned to the specified activity in OFSC</td>
<td>Retrieves all service requests assigned to the specified activity together with their properties</td>
<td></td>
</tr>
<tr>
<td>create_inventory_request</td>
<td>... create a service request assigned to a specific inventory in OFSC</td>
<td>Creates a service request with the specified properties assigned to the specified inventory</td>
<td></td>
</tr>
<tr>
<td>get_inventory_request_list</td>
<td>... retrieve data on all specified service requests assigned to the specified inventory in OFSC</td>
<td>Retrieves all service requests assigned to the specified inventory together with their properties</td>
<td></td>
</tr>
<tr>
<td>get_request</td>
<td>... retrieve properties for any specified request</td>
<td>Retrieves all properties of the specified service request</td>
<td></td>
</tr>
</tbody>
</table>
User Authentication Structure

All API methods use the 'user' structure as authentication to determine the permissions of the Oracle Field Service Cloud client company user.

All customers can use the Client_ID and Client_Secret instead of login and password to populate the User Authentication Structure with credentials:

1. Register an application.
   a. In the Field Service Cloud Manage interface, click Configuration and select Applications.
   b. In the left pane, click the plus icon to open the New application window, specify the Application Name and Application ID, and click Submit.
   c. Under Authentication settings, select the Authenticate using Client ID/Client Secret check box.
   d. Click Save.
2. Select the application and under Authentication settings section, click Show Client ID / Client secret to view the Client ID and Client Secret.

The following table describes the Oracle Field Service Cloud SOAP authentication structure mandatory fields.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>now</td>
<td>string</td>
<td>current time in ISO 8601 format</td>
</tr>
<tr>
<td>company</td>
<td>string</td>
<td>case-insensitive identifier of the Client for which data is to be retrieved. provided by Oracle during integration.</td>
</tr>
<tr>
<td>login</td>
<td>string</td>
<td>The client ID of the application.</td>
</tr>
</tbody>
</table>
| auth_string | string| authentication hash; The value of this field must be computed as follows:  
|           |       | auth_string = SHA256(now + SHA256(CLIENT_SECRET + SHA256(CLIENT_ID))); |

For example:

```xml
<user>
  <now>CURRENT_TIME</now>
  <login>CLIENT_ID</login>
  <company>INSTANCE_NAME</company>
  <auth_string>SHA256(CURRENT_TIME + SHA256(CLIENT_SECRET + SHA256(CLIENT_ID)))</auth_string>
</user>
```

Authentication

The 'user' structure is used for the request authentication. The relevant error is returned if the authentication fails.
### Number | Login | Description
---|---|---
1 | now | is different from the current time on the server and this difference exceeds the predefined time-window (30 minutes by default)
2 | company | cannot be found in the Oracle Field Service Cloud
3 | login | cannot be found for this company
4 | application | application is not authorized to use this API
5 | auth_string | when auth_string is not equal to: SHA256(now + SHA256(Client_Secret +SHA256(Client_ID)));;

Otherwise authentication is successful and the request is processed further.

### Detailed Methods Description

#### Route-related Methods

**'get_route' Method (Enhanced in API Version 2)**

The 'get_route' method is used to retrieve information on the specified resource's route for the specified day.

**'get_route' Request**

The 'get_route' method request specifies:

- the route for which data is to be retrieved
- properties to be retrieved for the route

The request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>date</td>
<td>Yes</td>
<td>string</td>
<td>date of the route for which data is to be retrieved in the YYYY-MM-DD format</td>
</tr>
<tr>
<td>resource_id</td>
<td>Yes</td>
<td>string</td>
<td>external ID of the resource whose route data is to be retrieved</td>
</tr>
<tr>
<td>property_filter</td>
<td>No</td>
<td>struct</td>
<td>each 'property_filter' element contains name of the activity property to be returned in the response</td>
</tr>
</tbody>
</table>
Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>property_filter</td>
<td>No</td>
<td>int</td>
<td>There can be any number of ‘property_filter’ elements. If empty or omitted, all available properties will be returned.</td>
</tr>
<tr>
<td>select_from</td>
<td>No</td>
<td>int</td>
<td>The number of the activity in the route starting from which the activities are to be selected (activities starting with ‘select_from’th are returned). If empty or omitted, all activities in the route and their properties are returned.</td>
</tr>
<tr>
<td>select_count</td>
<td>No</td>
<td>int</td>
<td>Total number of activities for which data is to be returned. If empty or omitted, properties for all activities in the route starting from ‘select_from’th will be returned.</td>
</tr>
</tbody>
</table>

• ‘get_route’ Request Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:toatech:ResourceManagement:1.0">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:get_route>
      <user>
        <now>2014-08-14T16:51:28Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>0e8c5b40e74532ea6e4a72cc819cdc9f</auth_string>
      </user>
      <date>2014-08-14</date>
      <resource_id>942495</resource_id>
      <property_filter>id</property_filter>
      <property_filter>appt_number</property_filter>
      <property_filter>time_slot</property_filter>
      <select_from>1</select_from>
      <select_count>2</select_count>
    </urn:get_route>
  </soapenv:Body>
</soapenv:Envelope>
```

‘get_route’ Response

The ‘get_route’ response contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>Transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>Transaction error description (if ‘result_code’ is other than ‘0’).</td>
</tr>
<tr>
<td>activity_list</td>
<td>struct</td>
<td>‘activity_list’ element</td>
</tr>
</tbody>
</table>
• 'activity_list' Element of 'get_route' response

'activity_list' contains all activity 'properties' specified in the 'property_filter' field of the request and available in OFSC for the activity for the number of activities in the route defined by 'select_count' starting from 'select_from'th. If 'property_filter' is not specified, all activity fields are returned.

A repeating/mass or shift related activity may be included in the calculation of the returned total and the activities section does not return any details. This happens as the activity has not been created and is not visible in the user interface. This applies to future dated activities where the repeating/mass or shift activity has not been created and is not visible in the user interface.

The 'activity_list' element contains the following:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>int</td>
<td>total amount of activities in the processed route</td>
</tr>
<tr>
<td>activities</td>
<td>struct</td>
<td>array of 'activity' elements, each being an array of 'properties' elements each containing one activity property</td>
</tr>
<tr>
<td>route_start_time</td>
<td>DateTime</td>
<td>time of the route start (YYYY-MM-DD HH:MM:SS)</td>
</tr>
<tr>
<td>route_end_time</td>
<td>DateTime</td>
<td>time of the route end (YYYY-MM-DD HH:MM:SS)</td>
</tr>
<tr>
<td>route_reactivation_time</td>
<td>DateTime</td>
<td>time of the route reactivation (YYYY-MM-DD HH:MM:SS) (available starting from API Version 2)</td>
</tr>
</tbody>
</table>

• 'get_route' response Example

```
  <SOAP-ENV:Body>
    <ns1:get_route_response>
      <result_code>0</result_code>
      <activity_list>
        <total>6</total>
        <activities>
          <activity>
            <properties>
              <name id="3998006">
                <value>test_get_route-ZEYVKEVUGE</value>
              </name>
            </properties>
          </activity>
          <activity>
            <properties>
              <name id="3998007">
                <value>test_get_route-GNLWSZYGFV</value>
              </name>
            </properties>
          </activity>
        </activities>
      </activity_list>
    </ns1:get_route_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'start_route' and 'end_route' methods

The 'start_route' method is used to activate or reactivate the resource's route.

The 'end_route' method is used to deactivate the resource's route.

Note: a route cannot be ended if there are pending/started activities/prework.

'start_route' and 'end_route' requests

Requests of 'start_route' and 'end_route' specify the route to be started (restarted) or ended and contain the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>resource_id</td>
<td>Yes</td>
<td>string</td>
<td>external ID of the resource whose route is to be started/restarted or ended</td>
</tr>
<tr>
<td>time</td>
<td>Yes</td>
<td>DateTime</td>
<td>time when the route is to be started/restarted or ended (YYYY-MM-DD HH:MM:SS)</td>
</tr>
<tr>
<td>date</td>
<td>No</td>
<td>date</td>
<td>date of the route in the YYYY-MM-DD format</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>default value: the resource's current date</td>
</tr>
</tbody>
</table>

• 'start_route' Request Example

```xml
  <SOAP-ENV:Body>
    <nsl:start_route>
      <user>
        <now>2014-08-14T16:49:43Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>dd83420944bcb9516a8d5def2cfa87d2</auth_string>
      </user>
      <resource_id>44030</resource_id>
      <time>2014-08-14 19:50:00</time>
      <date>2014-08-14</date>
    </nsl:start_route>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'end_route' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:end_route>
      <user>
        <now>2014-08-14T16:50:34Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>635baf3ea7c5be51259a79c0c11c2c91</auth_string>
      </user>
      <resource_id>Mister_NGXNYY</resource_id>
      <time>2014-08-14 23:51:00</time>
      <date>2014-08-14</date>
    </ns1:end_route>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelop>
```

'start_route' and 'end_route' responses

The 'start_route' and 'end_route' responses contain data on the method success/failure and consist of the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
</tbody>
</table>

'create_activity' Method

The 'create_activity' method is used to add a new activity to OFSC.
'create_activity' Request
The 'create_activity' method request specifies:

- properties to be set for the new activity
- date to be set for the new activity
- position in the route to be set for the activity

The request of 'create_activity' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>date</td>
<td>Yes</td>
<td>string</td>
<td>date of the route for which data is to be retrieved in the YYYY-MM-DD format</td>
</tr>
<tr>
<td>resource_id</td>
<td>Yes</td>
<td>string</td>
<td>external ID of the resource whose route data is to be retrieved</td>
</tr>
<tr>
<td>position_in_route</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity followed by the activity to be created valid values: any company-specific activity ID special keys: activity position is not changed created activity is not-ordered created activity is the first created activity is the last</td>
</tr>
<tr>
<td>property_filter</td>
<td>No</td>
<td>struct</td>
<td>each 'property_filter' element contains name of the activity property to be returned in the response there can be any number of 'property_filter' elements If empty or omitted, all available properties will be returned.</td>
</tr>
</tbody>
</table>

- 'properties' Array of 'create_activity' Request

The following activity properties are mandatory in the 'create_activity' request:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>worktype</td>
<td>Yes</td>
<td>enum</td>
<td>activity work type (if 'aworktype' is sent, it must contain the activity type ID, if 'worktype' is sent – the activity type label)</td>
</tr>
<tr>
<td>duration</td>
<td>Yes</td>
<td>int</td>
<td>length (duration) of the activity in minutes</td>
</tr>
</tbody>
</table>

Note: this parameter is mandatory for activity types with the 'Define duration manually' feature enabled.
### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>language</td>
<td>Yes</td>
<td>enum</td>
<td>notification language</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>company specific language label (en, es, etc.)</td>
</tr>
<tr>
<td>time_zone</td>
<td>Yes</td>
<td>enum</td>
<td>customer's time zone</td>
</tr>
<tr>
<td>team_id</td>
<td>Yes</td>
<td>string</td>
<td>external ID of the team-holder – the head resource within a team</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note: this parameter is mandatory for activity types with the &quot;Teamwork&quot; feature enabled.</td>
</tr>
</tbody>
</table>

- "create_activity" Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:create_activity>
      <user>
        <now>2014-08-14T16:52:21Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>e2a48e8863b134823d29a3cc866f9602</auth_string>
        </user>
        <date>2014-08-14</date>
        <resource_id>44030</resource_id>
        <position_in_route>first</position_in_route>
        <properties>
          <name>time_zone</name>
          <value>Eastern</value>
        </properties>
        <properties>
          <name>language</name>
          <value>en</value>
        </properties>
        <properties>
          <name>appt_number</name>
          <value>test_cancel_appointment-MOVTQIGRTQ</value>
        </properties>
        <properties>
          <name>customer_number</name>
          <value>PIVCDBASYZ</value>
        </properties>
        <properties>
          <name>name</name>
          <value>Mister Roboto</value>
        </properties>
        <properties>
          <name>zip</name>
          <value>12345</value>
        </properties>
        <properties>
          <name>aworktype</name>
          <value>33</value>
        </properties>
        <properties>
          <name>time_slot</name>
          <value>16-18</value>
        </properties>
    </ns1:create_activity>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'create_activity' Response
The 'create_activity' response contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>activity</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of activity properties for new activity</td>
</tr>
</tbody>
</table>

All available activity properties are returned.

'create_activity' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:create_activity_response>
      <result_code>0</result_code>
      <activity>
        <properties>
          <name>name</name>
          <value>Mister Roboto</value>
        </properties>
        <properties>
          <name>zip</name>
          <value>12345</value>
        </properties>
        <properties>
          <name>customer_number</name>
          <value>PIVCDBASYZ</value>
        </properties>
        <properties>
          <name>time_zone</name>
          <value>Eastern</value>
        </properties>
        <properties>
          <name>position_in_route</name>
          <value>1</value>
        </properties>
        <properties>
          <name>aworktype</name>
          <value>33</value>
        </properties>
        <properties>
          <name>time_slot</name>
          <value>16-18</value>
        </properties>
        <properties>
          <name>service_window_start</name>
          <value>16:00:00</value>
        </properties>
        <properties>
          <name>service_window_end</name>
          <value>18:00:00</value>
        </properties>
    </activity>
  </ns1:create_activity_response>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'start_activity' Method

The 'start_activity' method is used to set the activity status to 'started'.

'start_activity' Request

The 'start_activity' method request specifies:

- the activity to be started
- the activity properties to be updated
time of the operation

The 'start_activity' request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity to be started</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>for segmentable activities – segment ID</td>
</tr>
<tr>
<td>time</td>
<td>Yes</td>
<td>dateTime</td>
<td>activity start time in the YYYY-MM-DD HH:MM:SS format</td>
</tr>
<tr>
<td>date</td>
<td>No</td>
<td>date</td>
<td>date of the operation in the YYYY-MM-DD format</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>default value: current date of the resource</td>
</tr>
</tbody>
</table>

• 'start_activity' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:start_activity>
      <user>
        <now>2014-08-14T16:49:43Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>dd83420944bcb9516a8d5def2cfa87d2</auth_string>
      </user>
      <activity_id>3998006</activity_id>
      <date>2014-08-14</date>
      <time>2014-08-14 16:00:00</time>
    </ns1:start_activity>
  </SOAP-ENV:Body>
</soapenv:Envelope>
```

'start_activity' Response

The response of the 'start_activity' method contains data on the method success/failure and all available activity properties.

The response contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>activity</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of activity properties of the processed activity All available activity properties are returned.</td>
</tr>
</tbody>
</table>

• 'start_activity' response Example

```xml
  ...  
</soapenv:Envelope>
```
<SOAP-ENV:Body>
  <ns1:start_activity_response>
    <result_code>0</result_code>
    <activity>
      <properties>
        <name>name</name>
        <value>Mister Roboto</value>
      </properties>
      <properties>
        <name>zip</name>
        <value>12345</value>
      </properties>
      <properties>
        <name>customer_number</name>
        <value>JUVJWQVHMC</value>
      </properties>
      <properties>
        <name>time_zone</name>
        <value>Eastern</value>
      </properties>
      <properties>
        <name>type</name>
        <value>regular</value>
      </properties>
      <properties>
        <name>position_in_route</name>
        <value>1</value>
      </properties>
      <properties>
        <name>aworktype</name>
        <value>33</value>
      </properties>
      <properties>
        <name>time_slot</name>
        <value>16-18</value>
      </properties>
      <properties>
        <name>service_window_start</name>
        <value>16:00:00</value>
      </properties>
      <properties>
        <name>service_window_end</name>
        <value>18:00:00</value>
      </properties>
      <properties>
        <name>appt_number</name>
        <value>test_complete_appointment-XWHBQWRV1</value>
      </properties>
      <properties>
        <name>language</name>
        <value>en</value>
      </properties>
      <properties>
        <name>duration</name>
        <value>48</value>
      </properties>
      <properties>
        <name>start_time</name>
        <value>2014-08-14 16:00:00</value>
      </properties>
      <properties>
        <name>status</name>
        <value>started</value>
      </properties>
    </activity>
  </ns1:start_activity_response>
</SOAP-ENV:Body>
'complete_activity' Method

The 'complete_activity' method is used to set the activity status to 'completed'.

'complete_activity' Request

The 'complete_activity' method request specifies:

- the activity to be completed
- the activity properties to be updated
- time of the operation

The 'complete_activity' request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity to be started</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>for segmentable activities – segment ID</td>
</tr>
<tr>
<td>time</td>
<td>Yes</td>
<td>dateTime</td>
<td>activity end time in the YYYY-MM-DD HH:MM:SS format</td>
</tr>
<tr>
<td>date</td>
<td>No</td>
<td>date</td>
<td>date of the operation in the YYYY-MM-DD format</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>default value: current date of the resource</td>
</tr>
</tbody>
</table>
The Activity Management API allows completing a segment of a segmentable activity or the entire activity. To complete a segment, the user has to send a ‘complete_activity’ request as described above. The request can contain the optional ‘time_to_complete’ property adjusting the time remaining till the segmentable activity completion.

To complete the entire activity, the ‘complete_activity’ request must contain the ‘time_to_complete’ property with the value ‘0’, and the ‘multiday_activity_status’ property with the value ‘completed’. This request completes the selected segment simultaneously completing the entire segmentable activity.

• ‘complete_activity’ Request Example

```xml
 xmlns:ns1="urn:toa:activity">
 <SOAP-ENV:Body>
  <ns1:complete_activity>
   <user>
    <now>2014-08-14T16:49:43Z</now>
    <login>soap</login>
    <company>in132</company>
    <auth_string>dd83420944bcb9516a8d5def2cfa87d2</auth_string>
   </user>
   <activity_id>3998006</activity_id>
   <date>2014-08-14</date>
   <time>2014-08-14 16:10:00</time>
  </ns1:complete_activity>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

• ‘complete_activity’ Response

The response of the ‘complete_activity’ method contains data on the method success/failure and all available activity properties. The response contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if result_code is other than '0')</td>
</tr>
<tr>
<td>activity</td>
<td>struct</td>
<td>array of ‘properties’ elements each containing one of activity properties of the processed activity All available activity properties are returned.</td>
</tr>
</tbody>
</table>

• ‘complete_activity’ response Example

```xml
 xmlns:ns1="urn:toa:activity">
 <SOAP-ENV:Body>
  <ns1:complete_activity_response>
   <result_code>0</result_code>
   <activity>
    <properties>
     <name>Mister Roboto</name>
     <value>Mister Roboto</value>
    </properties>
   </activity>
  </ns1:complete_activity_response>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
<name>zip</name>
<value>12345</value>
</properties>

<name>customer_number</name>
<value>JUVJWQVHMC</value>
</properties>

<name>time_zone</name>
<value>Eastern</value>
</properties>

<name>type</name>
<value>regular</value>
</properties>

<name>position_in_route</name>
<value>1</value>
</properties>

<name>aworktype</name>
<value>33</value>
</properties>

<name>time_slot</name>
<value>16-18</value>
</properties>

<name>service_window_start</name>
<value>16:00:00</value>
</properties>

<name>service_window_end</name>
<value>18:00:00</value>
</properties>

<name>appt_number</name>
<value>test_complete_appointment-XWHBQVWRVB</value>
</properties>

<name>language</name>
<value>en</value>
</properties>

<name>start_time</name>
<value>2014-08-14 16:00:00</value>
</properties>

<name>status</name>
<value>complete</value>
</properties>

<name>id</name>
<value>3998006</value>
</properties>

<name>end_time</name>
<value>2014-08-14 16:00:00</value>
</properties>

<name>delivery_window_start</name>
<value>15:30:00</value>
</properties>

<name>delivery_window_end</name>
<value>21:00:00</value>
'cancel_activity' Method
The 'cancel_activity' method is used to set the status of a pending activity to 'cancelled' and of a started activity to 'notdone'.

'cancel_activity' Request
The 'cancel_activity' method request specifies:

- the activity to be canceled
- the activity properties to be updated
- time of the operation

The 'cancel_activity' request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity to be canceled</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>for segmentable activities – segment ID</td>
</tr>
<tr>
<td>time</td>
<td>Yes</td>
<td>dateTime</td>
<td>activity end time in the YYYY-MM-DD HH:MM:SS format</td>
</tr>
<tr>
<td>properties</td>
<td>No</td>
<td>array</td>
<td>array of 'properties' elements each containing one of activity properties to be updated for the processed activity</td>
</tr>
</tbody>
</table>

- Specifics of Segmentable Activities Cancelation
The Activity Management API allows canceling a segment of a segmentable activity or the entire activity. To cancel a segment, the user has to send a 'cancel_activity' request as described above. The request can contain the optional 'time_to_complete' property adjusting the time remaining till the segmentable activity completion.
To cancel the entire activity, the 'cancel_activity' request must contain the ‘time_to_complete’ property with the value ‘0’, and the ‘multiday_activity_status’ property with the value ‘cancelled’. This request cancels the selected segment simultaneously canceling the entire segmentable activity.

If the ‘cancel_activity’ request is sent for a started segment of a segmentable activity, both the segment and the entire activity can be set notdone. To set the status of only the started segment to ‘notdone’, the user has to send a ‘cancel_activity’ request as described above. The request can contain the optional ‘time_to_complete’ property adjusting the time remaining till the segmentable activity completion.

To set the entire activity notdone, the ‘cancel_activity’ request must be sent for a started segment and must contain the ‘time_to_complete’ property with the value ‘0’, and the ‘multiday_activity_status' property with the value 'notdone'.

- 'cancel_activity'Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:cancel_activity>
      <user>
        <now>2014-08-14T16:52:21Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>e2a48e8863b134823d29a3cc866f9602</auth_string>
      </user>
      <activity_id>3998010</activity_id>
      <date>2014-08-14</date>
      <time>2014-08-14 16:10:00</time>
      <properties>
        <name>new_property</name>
        <value>property</value>
      </properties>
    </ns1:cancel_activity>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- 'cancel_activity'Response

The response of the ‘cancel_activity' method contains data on the method success/failure and all available activity properties. The response contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>activity</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of activity properties of the processed activity All available activity properties are returned.</td>
</tr>
</tbody>
</table>

- 'cancel_activity' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:cancel_activity_response>
      <result_code>0</result_code>
      <activity>
        <properties>
          <name>name</name>
          <value>Mister Roboto</value>
        </properties>
      </activity>
    </ns1:cancel_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
</properties>

<properties>
    <name>zip</name>
    <value>12345</value>
</properties>

<properties>
    <name>customer_number</name>
    <value>PIVCDBASHY</value>
</properties>

<properties>
    <name>time_zone</name>
    <value>Eastern</value>
</properties>

<properties>
    <name>type</name>
    <value>regular</value>
</properties>

<properties>
    <name>aworktype</name>
    <value>33</value>
</properties>

<properties>
    <name>time_slot</name>
    <value>16-18</value>
</properties>

<properties>
    <name>service_window_start</name>
    <value>16:00:00</value>
</properties>

<properties>
    <name>service_window_end</name>
    <value>18:00:00</value>
</properties>

<properties>
    <name>appt_number</name>
    <value>test_cancel_appointment-MOVTQIGRTQ</value>
</properties>

<properties>
    <name>language</name>
    <value>en</value>
</properties>

<properties>
    <name>status</name>
    <value>cancelled</value>
</properties>

<properties>
    <name>id</name>
    <value>3998010</value>
</properties>

<properties>
    <name>delivery_window_start</name>
    <value>15:30:00</value>
</properties>

<properties>
    <name>delivery_window_end</name>
    <value>16:30:00</value>
</properties>

<properties>
    <name>date</name>
    <value>2014-08-14</value>
</properties>

<properties>
    <name>resource_id</name>
    <value>44030</value>
</properties>
'get_activity' Method
The 'get_activity' method is used to retrieve activity properties for the specified activity.

'get_activity' Request
The 'get_activity' method request specifies:
- activity to retrieve information for

The request of 'get_activity' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity data to be retrieved for</td>
</tr>
</tbody>
</table>

• 'get_activity' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_activity>
      <user>
        <now>2014-08-14T16:50:43Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>d0b15e0fb99d9fb1e67af7769293b6ff</auth_string>
      </user>
      <activity_id>3998006</activity_id>
    </ns1:get_activity>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'get_activity' Response
The 'get_activity' response contains data on the method success/failure and the properties of the activity specified in the request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>activity</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of activity properties for new activity All available activity properties are returned.</td>
</tr>
</tbody>
</table>
• 'get_activity' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_activity_response>
      <result_code>0</result_code>
      <activity>
        <name>name</name>
        <value>Mister Roboto</value>
      </activity>
      <properties>
        <name>zip</name>
        <value>12345</value>
      </properties>
      <properties>
        <name>customer_number</name>
        <value>LYKITMYMUN</value>
      </properties>
      <properties>
        <name>time_zone</name>
        <value>Eastern</value>
      </properties>
      <properties>
        <name>phone</name>
        <value>2325435</value>
      </properties>
      <properties>
        <name>email</name>
        <value>blarg@gmail.com</value>
      </properties>
      <properties>
        <name>type</name>
        <value>regular</value>
      </properties>
      <properties>
        <name>position_in_route</name>
        <value>1</value>
      </properties>
      <properties>
        <name>aworktype</name>
        <value>33</value>
      </properties>
      <properties>
        <name>time_slot</name>
        <value>16-18</value>
      </properties>
      <properties>
        <name>service_window_start</name>
        <value>16:00:00</value>
      </properties>
      <properties>
        <name>service_window_end</name>
        <value>18:00:00</value>
      </properties>
      <properties>
        <name>appt_number</name>
        <value>test_get_appointment-SNMJYRHXWT</value>
      </properties>
      <properties>
        <name>language</name>
        <value>en</value>
      </properties>
      <properties>
        <name>duration</name>
        <value>48</value>
      </properties>
    </ns1:get_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
</properties>
</activity>
</ns1:get_activity_response>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
**Note:** When activity notifications are delivered, the corresponding information is provided in the request response:
- delivery via 'cphone' (property label in Oracle Field Service Cloud) corresponds to 'phone_notification' in the response.
- delivery via 'cemail' (property label in Oracle Field Service Cloud) corresponds to 'email_notification' in the response.
- delivery via 'ccell' (property label in Oracle Field Service Cloud) corresponds to 'cell_notification' in the response.

Example Response:
```
<properties>
  <name>phone_notification</name>
  <value>1</value>
</properties>
<properties>
  <name>email_notification</name>
  <value>1</value>
</properties>
<properties>
  <name>cell_notification</name>
  <value>1</value>
</properties>
```

### 'get_activities' Method (Available Starting From API Version 2)

The method is available starting from API Version 2.

The 'get_activities' method is intended to return a list of activities matching certain filters. The distinctive feature of this method is that it allows searching activities by multiple criteria combining them on 'and' or 'or' principles.

The 'get_activities' request specifies the resource whose activities are to be returned (with the option of defining whether activities of child resources are to be returned), the date range in which the activities are to be searched and the filters by which the search results are to be selected. Also, the request can specify the properties to be returned for the found activities.

The request does not return any repeating/mass or shift activities unless the activity has been created and is visible in the user interface.

**Note:** processing of newly-created activities and changes to the existing activities takes certain time. For this reason, they may be missing in the search results, if the request is sent immediately after the activities creation or update. Newly-created activities and changes to the existing activities appear in the search results several seconds after the creation.

### 'get_activities' Request

The 'get_activities' method request specifies:

- resources whose activities are to be returned
- date range in which activities are to be searched
- filters which the search results are to match
- properties to be returned for the found activities

**Note:** the search is always case-insensitive.
The request of 'get_activities' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>resources</td>
<td>Yes</td>
<td>struct</td>
<td>array of 'resource' elements each defining a resource whose activities are to be returned</td>
</tr>
<tr>
<td>date_range</td>
<td>Yes</td>
<td>struct</td>
<td>element defining the interval of dates in which activities are to be searched</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Optionally, the request can be set to include non-scheduled activities.</td>
</tr>
<tr>
<td>select_from</td>
<td>No</td>
<td>int</td>
<td>the number of the activity starting from which the activities are to be selected (activities starting with 'select_from'th are returned)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>default value: '1'</td>
</tr>
<tr>
<td>select_count</td>
<td>Yes</td>
<td>int</td>
<td>total number of activities to be returned.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>No more records than the number defined in 'select_count' will be returned (but possibly less, if the query itself yields less results)</td>
</tr>
<tr>
<td>filter_expression</td>
<td>No</td>
<td>struct</td>
<td>element defining the filters which the activities must match to be included in the response</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If omitted, all activities in the specified date rage are returned.</td>
</tr>
<tr>
<td>required_properties</td>
<td>No</td>
<td>struct</td>
<td>array of 'label' elements each containing one of the activity properties to be returned for the found activities</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note: the 'position_in_route' field cannot be used in the 'get_activities' method.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If omitted, only the 'id' field will be returned.</td>
</tr>
</tbody>
</table>

- 'resource' Element of 'get_activities' Request

The 'resource' element specifies the resource whose activities are to be returned and contains the following parameters:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>Yes</td>
<td>string</td>
<td>external ID of the resource in whose activities the search is to be performed</td>
</tr>
</tbody>
</table>
• 'date_range' Element of 'get_activities' Request

The 'date_range' element defines the interval of dates in which activities are to be searched and contains the following parameters:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date_from</td>
<td>No</td>
<td>date</td>
<td>beginning of the date interval for the search in the YYYY-MM-DD format</td>
</tr>
<tr>
<td>date_to</td>
<td>No</td>
<td>date</td>
<td>end of the date interval for the search in the YYYY-MM-DD format</td>
</tr>
<tr>
<td>include_unscheduled</td>
<td>No</td>
<td>bool</td>
<td>flag defining whether unscheduled activities are to be returned in the response</td>
</tr>
</tbody>
</table>

valid values: true, false
default value: 'false'

Note: the 'date_range' element must always contain the 'date_from'/'date_to' parameters and/or the 'include_unscheduled' parameter. This way, activities will be searched in the specified period and/or in the unscheduled activities pool.

• 'filter_expression' Element of 'get_activities' Request

The 'filter_expression' element defines the criteria by which activities are to be searched. The 'filter_expression' element can represent a single expression and, in this case, the response will return activities matching only one field and its value. On the other hand, the 'filter_expression' can contain multiple expressions and define their combinations. In this case activities are compared to the entire group of expressions and only the activities matching all specified criteria are returned. The 'filter_expression' element defining a single expression contains the following parameters:
## Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>field</td>
<td>Yes</td>
<td>string</td>
<td>field or activity property used as the search criterion</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> the ‘position_in_route’ field cannot be used in the ‘get_activities’ method.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For company-defined properties, property labels are used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> file properties cannot be used in expressions.</td>
</tr>
<tr>
<td>operator</td>
<td>Yes</td>
<td>string</td>
<td>condition used to compare the activity property value with the value specified in the request</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>valid values:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• one_of – the property value is one of the array of values specified in the request</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• contains – the property value contains the specified substring</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• has_prefix – the property value has the specified prefix</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• has_suffix – the property value has the specified suffix</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• between – the property value is in the range between two specified values, both inclusive</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• null – the property specified in the request should be null to match the expression</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• equals – the property value is equal to the specified value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• less_than – the property value is less than the specified value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• greater_than – the property value is greater than the specified value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• less_than_or_equal – the property value is less than or equal to the specified value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• greater_than_or_equal – the property value is greater than or equal to the specified value</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> reverse operators are also possible for each of the above operators. For example: 'null' ≠ 'not_null', 'less_than' ≠ 'not_less_than', 'equals' ≠ 'not_equality', etc.)</td>
</tr>
<tr>
<td>value</td>
<td>Yes/No</td>
<td>string</td>
<td>value to which the activity property value is to be compared using the specified operator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mandatory for all operators except ‘null’. When the operator is ‘null’, no ‘value’ parameter is to be sent.</td>
</tr>
<tr>
<td>value</td>
<td>Yes/No</td>
<td>string</td>
<td>value to which the activity property value is to be compared using the specified operator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mandatory for all operators except ‘null’. When the operator is ‘null’, no ‘value’ parameter is to be sent.</td>
</tr>
</tbody>
</table>
The `filter_expression` element defining multiple expressions contains the following parameters:

- **group_operator**
  - **Required:** Yes
  - **Type:** string
  - **Description:** parameter defining the rule of comparing the activity property values to several expressions
    - **valid values:**
      - **or** – properties are compared to the expressions on the 'or' principle, i.e. the search result is to match either of the specified expressions
      - **and** – properties are compared to the expressions on the 'and' principle, i.e. the search result is to match all specified expressions joined by the 'and' operator

- **group**
  - **Required:** Yes
  - **Type:** struct
  - **Description:** array of `filter_expression` elements each defining a single expression to which the activity property is to be compared

If the 'group' structure contains multiple 'filter_expression' elements, the comparison is made to all of them on the 'or' or 'and' principle as defined by the 'group_operator'.

**Note:** values can be compared on several levels – that is, a 'filter_expression' element can include several 'filter_expression' elements joined by one group operator, while one or more of such second-level 'filter_expression' elements can include other 'filter_expression' elements joined by the same or different group operator.

In the following example, a single filter expression in which the activity status is to be 'pending' to match the search criterion is joined on the 'and' principle with another filter expression. Such second filter expression, in its turn, filters the activities with the 'type' equal to 'internal' or the 'xa_important' value equal to 'yes'. In simple words, such search will return activities of the 'pending' status and of the 'internal' type or activities of the 'pending' status with the 'xa_important' property set to 'yes'.

```xml
<filter_expression>
  <group_operator>and</group_operator>
  <group>
    <filter_expression>
      <field>status</field>  
      <operator>equals</operator> 
      <value>pending</value> 
    </filter_expression>
    <filter_expression>
      <group_operator>or</group_operator>
      <group>
        <expression>
          <field>type</field>
        </expression>
        <field>xa_important</field>
      </group>
    </filter_expression>
  </group>
</filter_expression>
```
• 'get_activities' Request Example

The following request searches activities assigned to 'Bucket A' and all its child resources in the date range from May 20, 2014 to May 22, 2014. The request is to return no more than 10 activities matching the following criteria: duration equal to 35 and traveling time equal to 15 or duration equal to 35 and name equal to 'testrest'. The following properties are to be returned for the found activities: 'id', 'date', 'resource_id', 'status', 'name', 'duration', 'traveling_time'.

```xml
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:urn="urn:toa:activity">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:get_activities>
      <user>
        <now>2014-05-22T08:30:44+0300</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>dce625f915d05f1163e40a24b0b068bedce625f915d05f1163e40a24b0b068be</auth_string>
      </user>
      <resources>
        <resource>
          <resource_id>Bucket A</resource_id>
          <include_children>all</include_children>
        </resource>
      </resources>
      <date_range>
        <date_from>2014-05-20</date_from>
        <date_to>2014-05-22</date_to>
      </date_range>
      <select_count>10</select_count>
      <filter_expression>
        <group_operator>and</group_operator>
        <group>
          <filter_expression>
            <field>duration</field>
            <operator>equals</operator>
            <value>35</value>
          </filter_expression>
          <filter_expression>
            <field>traveling_time</field>
            <operator>equals</operator>
            <value>15</value>
          </filter_expression>
        </group>
        <group_operator>or</group_operator>
        <group>
          <filter_expression>
            <field>name</field>
            <operator>equals</operator>
            <value>testrest</value>
          </filter_expression>
        </group>
      </filter_expression>
    </urn:get_activities>
  </soapenv:Body>
</soapenv:Envelope>
```
Oracle Field Service Cloud
Integrating with Activity Management API

Chapter 2
Activity Management API Overview

```
<urn:get_activities>
<soapenv:Body>
</urn:get_activities>
</soapenv:Envelope>
```

'get_activities' Response
The 'get_activities' response contains data on the method success/failure as well as the list of activities matching the search criteria with their properties. Also, the response includes the filter expressions used in the search in human-readable format and the query statistics (execution time, memory usage, etc.)

The 'get_activities' response contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>expression</td>
<td>struct</td>
<td>expression tree containing the filter expressions used in the request but converted to easily readable format. The expression tree is used to verify that the search has been set up correctly.</td>
</tr>
<tr>
<td>activities</td>
<td>struct</td>
<td>array of 'activity' elements each containing properties of one of the activities returned by the search</td>
</tr>
<tr>
<td>have_more_results</td>
<td>bool</td>
<td>parameter indicating that the method returned less matching results than the number defined in the 'select_count' field. The results number may be restricted because the response size or time limit have been exceeded. In this case the 'statistics' element will have a corresponding text warning. Note: the 'have_more_results' parameter has only one valid value – 'true'. Otherwise, the element is not returned.</td>
</tr>
<tr>
<td>statistics</td>
<td>struct</td>
<td>element containing the query metrics (for example, execution time, memory usage, etc.) to be used in load estimation and troubleshooting</td>
</tr>
</tbody>
</table>

- 'expression' Element of 'get_activities' response
The 'expression' element of the 'get_activities' response contains the same filter expressions which were sent in the request to filter the list of found activities. The filter expressions used in the 'get_activities' request above are interpreted in the following 'expression' element:

```
<expression>
  ( 'duration' equals 35
  and
  ( 'traveling_time' equals 15
  or
    'name' equals testrest )
)</expression>
```

The user can easily see the activities search criteria and verify whether they are correct and actually aimed at finding the activities the user needs.

- 'get_activities' response Example

```
  <soapenv:Body>
    <urn:get_activities_response>
      <result_code>0</result_code>
      <expression>
        'duration' equals 35
        and
        ( 'traveling_time' equals 15
        or
          'name' equals testrest
        )
      </expression>
      <activities>
        <activity>
          <properties>
            <name>id</name>
            <value>23225242</value>
          </properties>
          <properties>
            <name>date</name>
            <value>2014-05-20</value>
          </properties>
          <properties>
            <name>resource_id</name>
            <value>Tech 1</value>
          </properties>
          <properties>
            <name>status</name>
            <value>pending</value>
          </properties>
          <properties>
            <name>name</name>
            <value>testrest</value>
          </properties>
          <properties>
            <name>duration</name>
            <value>35</value>
          </properties>
          <properties>
            <name>traveling_time</name>
            <value>15</value>
          </properties>
        </activity>
        <activity>
          <properties>
            <name>id</name>
            <value>23225237</value>
          </properties>
        </activity>
      </activities>
    </urn:get_activities_response>
  </soapenv:Body>
</soapenv:Envelope>
```
<activity>
    <properties>
        <name>id</name>
        <value>23225241</value>
    </properties>
    <properties>
        <name>date</name>
        <value>2014-05-21</value>
    </properties>
    <properties>
        <name>resource_id</name>
        <value>Tech 3</value>
    </properties>
    <properties>
        <name>status</name>
        <value>pending</value>
    </properties>
    <properties>
        <name>name</name>
        <value>testrest</value>
    </properties>
    <properties>
        <name>duration</name>
        <value>35</value>
    </properties>
    <properties>
        <name>traveling_time</name>
        <value>15</value>
    </properties>
</activity>

<activities>
</activities>

<statistics>
Query time: 0.0115
Processing time: 0.0003
Results returned: 3
Properties per result: 7
Response size: 0.002 MB
Peak memory: 1.750 MB
Queue size: 1 Queue wait time: 0
</statistics>
</urn:get_activities_response>
</soapenv:Body>
Restrictions
The 'get_activities' request and response are subject to certain restrictions preventing the system overload:

- **Requests Number Restriction**
  - no more than 3 'get_activities' requests can be run simultaneously. If more than 3 requests are sent, the rest of the requests are queued for execution
  - no more than 10 requests can be queued. If 10 requests are queued already and more requests are sent, they are immediately rejected with the corresponding error message
  - maximum time of a request waiting for execution in the queue is 30 seconds. When the waiting time is exceeded, the corresponding error message is returned

- **Activities Number Restriction**
  - No more than 100 000 activities can be returned in a single response. If the 'select_count' parameter in the request is set to a value greater than 100 000, the response will contain an error message with the explanation of the error.

- **Date Range Restriction**
  - No more than 31 days can be searched in a single request. If the date range ('date_to' – 'date_from') is longer than 31 days, the response will contain an error message with the explanation of the error.

- **Response Size Restriction**
  - A single response size is limited to 20 MB. If the response size is close to 20 MB, the method returns fewer activities than requested and the response contains the 'have_more_results' parameter set to 'true'. This parameter indicates that there are more activities to be returned. In this case a new request with 'select_from' set to the last returned count + 1 is to be sent.

- **Operation Run Time Restriction**
  - A single operation can run no longer than 30 seconds. When the 30-second time limit has been reached, the method returns fewer activities than requested and the response contains the 'have_more_results' parameter set to 'true'. This parameter indicates that there are more activities to be returned. In this case a new request with 'select_from' set to the last returned count + 1 is to be sent. If the time limit has been exceeded before any activities are included in the response (for example, due to slow database response) an error is returned.

- **Company-defined Properties Number Restriction**
  - The maximum number of company-defined properties used in the request and/or returned in the response is 50. If more than 50 properties are included in the request, the response will contain an error message with the explanation of the error. The 50-property restriction includes both the company-defined properties used in the 'required_properties' list and in the filter expressions.

'get_multiday_activities' Method (Available Starting from API Version 3)
The method is available starting from API Version 3.
The 'get_multiday_activities' method is used to retrieve the list of all segmentable activities assigned to the specified resource and, optionally, their properties.

'get_multiday_activities' Request
The 'get_multiday_activities' method request specifies:

- the resource for which segmentable activities are to be returned
• optionally, the properties to be returned for each activity

The request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>‘user’ structure</td>
</tr>
<tr>
<td>resource_id</td>
<td>Yes</td>
<td>string</td>
<td>external ID of the resource for which segmentable activity list is to be retrieved</td>
</tr>
<tr>
<td>required_properties</td>
<td>No</td>
<td>struct</td>
<td>each ‘required_properties’ element contains the name of the activity property to be returned in the response. There can be any number of ‘required_properties’ elements. If omitted, only IDs of the segmentable activities will be returned.</td>
</tr>
</tbody>
</table>

• ‘get_multiday_activities’ Request Example

```xml
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:get_multiday_activities>
      <user>
        <now>2015-05-18T22:02:13Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>b07ceff067beb7c628985131c1464148</auth_string>
      </user>
      <resource_id>11102</resource_id>
    </ns1:get_multiday_activities>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

• ‘get_multiday_activities’ Response

The ‘get_multiday_activities’ response contains data on the method success/failure, as well as the list of the segmentable activities assigned to the specified resource and, optionally, their properties.

The ‘get_multiday_activities’ response contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if ‘result_code’ is other than ‘0’)</td>
</tr>
<tr>
<td>activities</td>
<td>struct</td>
<td>array of ‘activity’ elements each containing properties of one of the segmentable activities returned by the search</td>
</tr>
</tbody>
</table>

• ‘get_multiday_activities’ response Example
'get_multiday_activity_parts' Method (Available Starting from API Version 3)

The method is available starting from API Version 3.

The 'get_multiday_activity_parts' method is used to retrieve the list of all segments of the specified segmentable activity and optionally, their properties. The method also returns the data of the segmentable activity progress.

'get_multiday_activity_parts' Request

The 'get_multiday_activity_parts' method request specifies:

- the activity for which segment data is to be returned
- optionally, the properties to be returned for each segment

The request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the segmentable activity for which the list of segments is to be retrieved</td>
</tr>
</tbody>
</table>
| required_properties| No       | struct | each 'required_properties' element contains the name of the activity property to be returned in the response  
|                    |          |        | there can be any number of 'required_properties' elements                  |
|                    |          |        | If omitted, only IDs of the segmentable activity segments will be returned. |

'get_multiday_activity_parts' Request Example

```xml
  <SOAP-ENV:Body>
    <nsl:get_multiday_activity_parts>
      <user>
        <now>2015-03-05T17:05:50Z</now>
        <login>soap</login>
        <company>in132</company>
      </user>
    </nsl:get_multiday_activity_parts>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Chapter 2
Activity Management API Overview

The `get_multiday_activity_parts` response contains data on the method success/failure, the data of the segmentable activity progress as well as the list of the specified segmentable activity segments with their properties.

The `get_multiday_activity_parts` response contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if <code>result_code</code> is other than '0')</td>
</tr>
<tr>
<td>multiday_info</td>
<td>struct</td>
<td>array of <code>properties</code> elements each containing one of the segmentable activity properties</td>
</tr>
<tr>
<td>activity_list</td>
<td>struct</td>
<td>'activity_list' element</td>
</tr>
</tbody>
</table>

- **'multiday_info' Element of 'get_multiday_activity_parts' response**

The 'multiday_info' element of the `get_multiday_activity_parts` response contains the properties describing the progress of the segmentable activity. It always consists of the following properties:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>total_duration</td>
<td>int</td>
<td>total duration of the segmentable activity in minutes</td>
</tr>
<tr>
<td>started_duration</td>
<td>int</td>
<td>cumulative duration of all started segments of the segmentable activity</td>
</tr>
<tr>
<td>completed_duration</td>
<td>int</td>
<td>cumulative duration of all completed and not done segments of the segmentable activity</td>
</tr>
<tr>
<td>total_count</td>
<td>int</td>
<td>total number of segments of the segmentable activity</td>
</tr>
<tr>
<td>completed_count</td>
<td>int</td>
<td>total number of completed segments of the segmentable activity</td>
</tr>
</tbody>
</table>

- **'get_multiday_activity_parts' response Example**

```xml
  <SOAP-ENV:Body>
    <nsl:get_multiday_activity_parts_response>
      ...<auth_string>d344f30186650e2e549e94f5d5b51822</auth_string>
    </nsl:get_multiday_activity_parts_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
<result_code>0</result_code>
<multiday_info>
  <properties>
    <name>total_duration</name>
    <value>6000</value>
  </properties>
  <properties>
    <name>started_duration</name>
    <value>0</value>
  </properties>
  <properties>
    <name>completed_duration</name>
    <value>0</value>
  </properties>
  <properties>
    <name>total_count</name>
    <value>15</value>
  </properties>
  <properties>
    <name>completed_count</name>
    <value>0</value>
  </properties>
</multiday_info>
<multiday_parts>
  <activity>
    <properties>
      <name>id</name>
      <value>3998007</value>
    </properties>
    <properties>
      <name>date</name>
      <value>2015-03-06</value>
    </properties>
    <properties>
      <name>resource_id</name>
      <value>44030</value>
    </properties>
    <properties>
      <name>status</name>
      <value>pending</value>
    </properties>
    <properties>
      <name>type</name>
      <value>multiday_activity_segment</value>
    </properties>
  </activity>
  <activity>
    <properties>
      <name>id</name>
      <value>3998008</value>
    </properties>
    <properties>
      <name>date</name>
      <value>2015-03-07</value>
    </properties>
    <properties>
      <name>resource_id</name>
      <value>44030</value>
    </properties>
    <properties>
      <name>status</name>
      <value>pending</value>
    </properties>
    <properties>
      <name>type</name>
      <value>multiday_activity_segment</value>
    </properties>
  </activity>
</multiday_parts>
Oracle Field Service Cloud
Integrating with Activity Management API

Chapter 2
Activity Management API Overview

'get_activity_work_skills' Method

The 'get_activity_work_skills' method is used to retrieve the list of work skills of the specified activity. The method returns the list of work skills and their required and preferable qualification level for the activity specified in the request as well as the list of its capacity categories. Activity belongs to a capacity category if all work skills of the capacity category make a subset of the skills calculated for the activity. A work skill of capacity category matches an activity work skill if the qualification level defined for the skill in the capacity category is not higher than the qualification level defined as required for the activity.

'get_activity_work_skills' Request

The 'get_activity_work_skills' method request specifies:

- activity for which work skills are to be retrieved

The request of 'get_activity_work_skills' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity for which work skills are to be retrieved</td>
</tr>
</tbody>
</table>

'get_activity_work_skills' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_activity_work_skills>
      <user>
        <now>2014-08-14T16:53:39Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>2034a8829f0ab5c1bf409338e3338aa6</auth_string>
      </user>
      <activity_id>3998006</activity_id>
    </ns1:get_activity_work_skills>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'get_activity_work_skills' Response

The 'get_activity_work_skills' response contains the following elements:

- list of work skills of the activity
- list of capacity categories of the activity
- required and preferable qualification level of the work skill

The response of 'get_activity_work_skills' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>work_skill</td>
<td>struct</td>
<td>array of 'work_skill' elements</td>
</tr>
<tr>
<td>capacity_category</td>
<td>struct</td>
<td>capacity category the activity belongs to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>activity belongs to a capacity category if all work skills of the capacity category make a subset of the skills calculated for the activity</td>
</tr>
</tbody>
</table>

- 'work_skill' Element of 'get_activity_work_skills' response

The 'work_skill' element contains details of the activity work skills and can contain the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>enum</td>
<td>label of the work skill assigned to the activity</td>
</tr>
<tr>
<td>description</td>
<td>enum</td>
<td>name of the work skill</td>
</tr>
<tr>
<td>required_level</td>
<td>int</td>
<td>required level of the work skill</td>
</tr>
<tr>
<td>preferable_level</td>
<td>int</td>
<td>preferable level of the work skill</td>
</tr>
</tbody>
</table>

- 'capacity_category' Element of 'get_activity_work_skills' response

The 'capacity_category' element contains details of the capacity categories to which the specified activity belongs. It can contain the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>enum</td>
<td>label of the capacity category</td>
</tr>
</tbody>
</table>
Oracle Field Service Cloud
Integrating with Activity Management API

Chapter 2
Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>enum</td>
<td>name of the capacity category</td>
</tr>
</tbody>
</table>

- 'get_activity_work_skills' response Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <SOAP-ENV:Body>
    <ns1:get_activity_work_skills_response>
      <result_code>0</result_code>
      <work_skill>
        <name>IN</name>
        <description>Install</description>
        <required_level>20</required_level>
        <preferable_level>60</preferable_level>
      </work_skill>
      <work_skill>
        <name>UP</name>
        <description>Upgrade</description>
        <required_level>1</required_level>
        <preferable_level>1</preferable_level>
      </work_skill>
      <capacity_category>
        <name>CAP</name>
        <description>CAP</description>
      </capacity_category>
    </ns1:get_activity_work_skills_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'update_activity' and 'reopen_activity' Methods

The 'update_activity' method is used to modify the properties of the specified activity.

The 'reopen_activity' method is used to create a 'pending' activity duplicating the specified 'cancelled', 'notdone' or 'completed' activity and update such pending activity.

'update_activity' and 'reopen_activity' Request

The requests of 'reopen_activity' and 'update_activity' methods specify:

- activity to be processed
- properties to be updated ('update_activity' – for the activity specified and 'reopen_activity' – for the pending activity duplicating the processed one)
- activity position in the route ('update_activity' – for the activity specified and 'reopen_activity' – for the pending activity duplicating the processed one)

Requests of both methods contain the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity to be updated/reopened</td>
</tr>
</tbody>
</table>
### Activity Management API Overview

**Name** | **Required** | **Type** | **Description**
--- | --- | --- | ---
position_in_route | Yes | string | ID of the activity followed by the processed activity. Valid values: any company-specific activity ID. Special keys:
- activity position is not changed
- created activity is not-ordered
- created activity is the first
- created activity is the last

properties | No | array | array of ‘properties’ elements each containing one of activity properties to be set for updated/reopened activity. All properties are optional for the method.

---

**'update_activity' Request Example**

```xml
  <SOAP-ENV:Body>
    <ns1:update_activity>
      <user>
        <now>2014-08-14T16:54:24Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>baf0616de48063b8c62fe2ec03a31232</auth_string>
      </user>
      <activity_id>3998007</activity_id>
      <position_in_route>unchanged</position_in_route>
      <properties>
        <name>time_of_booking</name>
        <value>2014-08-14 01:01:02</value>
      </properties>
    </ns1:update_activity>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**'reopen_activity' Request Example**

```xml
  <SOAP-ENV:Body>
    <ns1:reopen_activity>
      <user>
        <now>2014-08-21T12:19:10Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>9dfe777259a7ca4454ed01cd63af0b78</auth_string>
      </user>
      <activity_id>3998006</activity_id>
      <position_in_route>first</position_in_route>
      <properties>
        <name>name</name>
        <value>New Mister Roboto</value>
      </properties>
    </ns1:reopen_activity>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'update_activity' and 'reopen_activity' Responses

The 'update_activity' and 'reopen_activity' responses contain data on the method success/failure and all properties of the processed activity. The responses of 'update_activity' and 'reopen_activity' methods contain the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>activity</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of activity properties to be set for updated/reopened activity. All available activity properties are returned.</td>
</tr>
</tbody>
</table>

'update_activity' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:update_activity_response>
      <result_code>0</result_code>
      <activity>
        <properties>
          <name>time_of_booking</name>
          <value>2014-08-14 01:01:02</value>
        </properties>
        <properties>
          <name>name</name>
          <value>Mister Roboto</value>
        </properties>
        <properties>
          <name>zip</name>
          <value>12345</value>
        </properties>
        <properties>
          <name>customer_number</name>
          <value>YVXADTCZWQ</value>
        </properties>
        <properties>
          <name>time_zone</name>
          <value>Eastern</value>
        </properties>
        <properties>
          <name>type</name>
          <value>regular</value>
        </properties>
        <properties>
          <name>position_in_route</name>
          <value>2</value>
        </properties>
        <properties>
          <name>aworktype</name>
          <value>33</value>
        </properties>
        <properties>
          <name>time_slot</name>
          <value>16-18</value>
        </properties>
        <properties>
          <name>service_window_start</name>
          <value>16:00:00</value>
        </properties>
      </activity>
    </ns1:update_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
• 'reopen_activity' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:reopen_activity_response>
      <result_code>0</result_code>
      <activity>
        <properties>
          <name>id</name>
          <value>3998007</value>
        </properties>
        <properties>
          <name>end_time</name>
          <value>2014-08-14 18:06:00</value>
        </properties>
        <properties>
          <name>delivery_window_end</name>
          <value>18:00:00</value>
        </properties>
        <properties>
          <name>traveling_time</name>
          <value>30</value>
        </properties>
        <properties>
          <name>date</name>
          <value>2014-08-14</value>
        </properties>
        <properties>
          <name>resource_id</name>
          <value>44030</value>
        </properties>
      </activity>
    </ns1:reopen_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
<name>name</name>
<value>New Mister Roboto</value>
</properties>
<properties>
<name>zip</name>
<value>12345</value>
</properties>
<properties>
<name>customer_number</name>
<value>IUAGMXQFTU</value>
</properties>
<properties>
<name>time_zone</name>
<value>Eastern</value>
</properties>
<properties>
<name>type</name>
<value>reopened</value>
</properties>
<properties>
<name>position_in_route</name>
<value>1</value>
</properties>
<properties>
<name>aworktype</name>
<value>33</value>
</properties>
<properties>
<name>time_slot</name>
<value>16-18</value>
</properties>
<properties>
<name>service_window_start</name>
<value>16:00:00</value>
</properties>
<properties>
<name>service_window_end</name>
<value>18:00:00</value>
</properties>
<properties>
<name>appt_number</name>
<value>test_reopen_appointment-DGGSDVERSR</value>
</properties>
<properties>
<name>language</name>
<value>en</value>
</properties>
<properties>
<name>duration</name>
<value>48</value>
</properties>
<properties>
<name>start_time</name>
<value>2014-08-21 16:00:00</value>
</properties>
<properties>
<name>status</name>
<value>pending</value>
</properties>
<properties>
<name>id</name>
<value>3998008</value>
</properties>
<properties>
<name>end_time</name>
<value>2014-08-21 16:48:00</value>
'search_activities' Method

The method enables retrieving the list of activities with the specified value in the specified field for the specified time period.

'search_activities' Request

The 'search_activities' method request specifies:

- time period to search for activity in
- activity property value to search for
- specific number of activities to return from the list of activities found
- the way activities should be ordered in the response
- properties to be retrieved for the specified activity

The request of 'search_activities' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>search_in</td>
<td>Yes</td>
<td>string</td>
<td>activity field or property in which the value defined with 'search_for' is to be searched</td>
</tr>
<tr>
<td>search_for</td>
<td>Yes</td>
<td>string</td>
<td>value to search for</td>
</tr>
</tbody>
</table>

The search is performed in the activity properties specified in OFSC (Manage & Company Settings & Business Rules & Activity Search Fields). Search Fields can be selected from the available fields (caddress, ccell, cphone, cemail, cname, appt_number, customer_number, czip) and the properties defined for the activity.
### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date_from</td>
<td>Yes</td>
<td>date</td>
<td>beginning of the date interval for the search in the YYYY-MM-DD format</td>
</tr>
<tr>
<td>date_to</td>
<td>Yes</td>
<td>date</td>
<td>end of the date interval for the search in the YYYY-MM-DD format</td>
</tr>
<tr>
<td>select_from</td>
<td>Yes</td>
<td>int</td>
<td>the number of the activity starting from which the activities are to be selected (activities starting with 'select_from'th are returned)</td>
</tr>
<tr>
<td>select_count</td>
<td>Yes</td>
<td>int</td>
<td>total number of activities to be returned. No more records than the number defined in 'select_count' will be returned (but possibly less, if the query itself yields fewer rows)</td>
</tr>
<tr>
<td>order</td>
<td>No</td>
<td>enum</td>
<td>enum value defining the order of the activities found valid values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• asc – ascending</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• desc – descending</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>default value: ‘desc’</td>
</tr>
<tr>
<td>property_filter</td>
<td>No</td>
<td>array</td>
<td>array of 'properties' elements each containing one of activity properties to be returned for the found activities if omitted, ‘id’, ‘resource_id’ and ‘date’ will be returned for all found activities, where available</td>
</tr>
</tbody>
</table>

- **'search_activities' Request Example**

  ```xml
  <SOAP-ENV:Body>
  <ns1:search_activities>
    <user>
      <now>2014-08-14T16:51:53Z</now>
      <login>soap</login>
      <company>in132</company>
      <auth_string>1ff9b3c9461ccff5936845c7c65232b74</auth_string>
    </user>
    <search_in>appt_number</search_in>
    <search_for>test_sea</search_for>
    <date_from>2014-08-13</date_from>
    <date_to>2014-08-15</date_to>
    <select_from>1</select_from>
    <select_count>1000</select_count>
    <order>desc</order>
  </ns1:search_activities>
  </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
  ```

**'search_activities' Response**

Response of 'search_activities' contains data on the method success/failure and the list of the found activities with their properties.
## Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>activity_list</td>
<td>struct</td>
<td>list</td>
</tr>
</tbody>
</table>

- **'activity_list' Element of 'search_activities' response**

The 'activity_list' element contains the total number of activities matching the search criteria and the properties of such activities.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>int</td>
<td>number of found activities</td>
</tr>
<tr>
<td>activities</td>
<td>array</td>
<td>array of 'activity' elements each containing the properties specified in the 'property_filter' for one activity matching the search criteria</td>
</tr>
</tbody>
</table>

if 'property_filter' is empty, 'id', 'resource_id' and 'date' will be returned for all found activities, where a

- **'search_activities' response Example**

```xml
  <SOAP-ENV:Body>
    <ns1:search_activities_response>
      <result_code>0</result_code>
      <activity_list>
        <total>2</total>
        <activities>
          <activity>
            <properties>
              <name>id</name>
              <value>3998006</value>
            </properties>
            <properties>
              <name>resource_id</name>
              <value>44030</value>
            </properties>
            <properties>
              <name>date</name>
              <value>2014-08-14</value>
            </properties>
          </activity>
          <activity>
            <properties>
              <name>id</name>
              <value>3998007</value>
            </properties>
            <properties>
              <name>resource_id</name>
              <value>44031</value>
            </properties>
          </activity>
        </activities>
      </activity_list>
    </ns1:search_activities_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'set_resource_preferences' Method

This method allows defining the resources preferred, required or forbidden for the activity to implement the company-specific business logics. If any resource preferences already exist for the specified activity, the 'set_resource_preferences' method updates them according to the request.

'set_resource_preferences' Request

The 'set_resource_preferences' method request specifies:

- activity the preferences are set for
- resources required, preferred and forbidden for the specified activity

The request of 'set_resource_preferences' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity for which preferences are to be set</td>
</tr>
<tr>
<td>preference</td>
<td>No</td>
<td>struct</td>
<td>preference type defined for the activity</td>
</tr>
</tbody>
</table>

There can be as many 'preference' elements, as many preferences should be defined for the activity.

If omitted, all existing resource preferences set for the activity will be deleted.

- 'preference' Element of 'set_resource_preferences' Request

The 'preference' element contains the preferences set for the resource identified by the external ID.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>Yes</td>
<td>string</td>
<td>external ID of the resource for which the preference is set</td>
</tr>
<tr>
<td>type</td>
<td>Yes</td>
<td>enum</td>
<td>type of preference: valid values:</td>
</tr>
</tbody>
</table>

- required – if any of the resources in the 'preference' element has 'type' set to 'required', only one of such resources can be assigned the activity.
Chapter 2
Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>preferred</td>
<td></td>
<td></td>
<td>- if no resources in the 'preference' element have 'type' set to 'required', the resources with 'type' set to 'preferred' will have the priority when the activity is assigned</td>
</tr>
<tr>
<td>forbidden</td>
<td></td>
<td></td>
<td>- forbids resources with 'type' set to 'forbidden' cannot be assigned the activity</td>
</tr>
</tbody>
</table>

'Set_resource_preferences' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:set_resource_preferences>
      <user>
        <now>2014-08-14T16:50:17Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>97ca46f291eb3884b7e126a67a522ce1</auth_string>
      </user>
      <activity_id>3998006</activity_id>
      <preference>
        <resource_id>660151</resource_id>
        <type>forbidden</type>
      </preference>
      <preference>
        <resource_id>44035</resource_id>
        <type>preferred</type>
      </preference>
    </ns1:set_resource_preferences>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'Set_resource_preferences' Response

The 'set_resource_preferences' response contains data on the method success/failure.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
</tbody>
</table>

'Set_resource_preferences' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:set_resource_preferences_response>
      <result_code>0</result_code>
    </ns1:set_resource_preferences_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'Get_resource_preferences' Method

This method allows retrieving data on the resources preferred, required or forbidden for the activity.
'get_resource_preferences' Request
The 'get_resource_preferences' method request specifies:

- activity for which the preferences are to be retrieved

The 'set_resource_preferences' request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity for which the preferences are to be retrieved</td>
</tr>
</tbody>
</table>

'get_resource_preferences' Request Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:urn="urn:toatech:ResourceManagement:1.0">
    <SOAP-ENV:Body>
        <ns1:get_resource_preferences>
            <user>
                <now>2014-08-14T16:50:17Z</now>
                <login>soap</login>
                <company>in132</company>
                <auth_string>97ca46f291eb3884b7e126a67a522ce1</auth_string>
            </user>
            <activity_id>3998006</activity_id>
        </ns1:get_resource_preferences>
    </SOAP-ENV:Body>
</soapenv:Envelope>
```

'get_resource_preferences' Response
The 'get_resource_preferences' response contains data on the method success/failure and data on the resources preferred, required or forbidden for the activity.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>preference</td>
<td>struct</td>
<td>preference type defined for the user found for the activity</td>
</tr>
</tbody>
</table>

The 'preference' element contains the preferences set for the resource identified by the external ID.
### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>string</td>
<td>external ID of the resource for which the preference is set</td>
</tr>
<tr>
<td>type</td>
<td>enum</td>
<td>type of preference:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>valid values:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• required – if any of the resources in the ‘preference’ element has ‘type’ set to ‘required’, only one of such resources can be assigned the activity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• preferred – if no resources in the ‘preference’ element have ‘type’ set to ‘required’, the resources with ‘type’ set to ‘preferred’ will have the priority when the activity is assigned</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• forbidden – resources with ‘type’ set to ‘forbidden’ cannot be assigned the activity</td>
</tr>
</tbody>
</table>

- ‘get_resource_preferences’ response Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_resource_preferences_response>
      <result_code>0</result_code>
      <preference>
        <resource_id>44035</resource_id>
        <type>preferred</type>
      </preference>
      <preference>
        <resource_id>660151</resource_id>
        <type>forbidden</type>
      </preference>
    </ns1:get_resource_preferences_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 'delay_activity' Method

The ‘delay_activity’ method changes the ‘end_time’ property of the activity and can update other specified activity properties.

#### 'delay_activity' Request

The ‘delay_activity’ method request specifies:

- activity to be processed
- properties to be updated
- activity delay period

The ‘delay_activity’ request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>‘user’ structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity to be processed</td>
</tr>
<tr>
<td>value</td>
<td>Yes</td>
<td>int</td>
<td>delay in minutes</td>
</tr>
</tbody>
</table>
Chapter 2
Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>No</td>
<td>date</td>
<td>date of the operation in the YYYY-MM-DD format default value: current date for the resource</td>
</tr>
<tr>
<td>properties</td>
<td>No</td>
<td>array</td>
<td>array of 'properties' elements each containing one of activity properties to be updated for the processed activity If omitted, no activity properties are updated</td>
</tr>
</tbody>
</table>

- 'delay_activity' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:delay_activity>
      <user>
        <now>2014-08-14T16:50:25Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>0a9505245bfa5102f6d9f321d1db7f4b</auth_string>
      </user>
      <activity_id>3998006</activity_id>
      <value>30</value>
      <properties>
        <name>act_property</name>
        <value>new property</value>
      </properties>
    </ns1:delay_activity>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- 'delay_activity' Response
The 'delay_activity' response contains data on the method success/failure and properties of the activity.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>activity</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of activity properties of the processed (updated) activity</td>
</tr>
</tbody>
</table>

  All available activity properties are returned.

- 'delay_activity' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:delay_activity_response>
      <result_code>0</result_code>
      <activity>
        <properties>
          <name>name</name>
          <value>Mister Roboto</value>
        </properties>
      </activity>
    </ns1:delay_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
<name>zip</name><value>12345</value></properties>
<properties>
<name>customer_number</name><value>VRHGHAXSQT</value></properties>
<properties>
<name>time_zone</name><value>Eastern</value></properties>
<properties>
<name>type</name><value>regular</value></properties>
<properties>
<name>position_in_route</name><value>1</value></properties>
<properties>
<name>aworktype</name><value>33</value></properties>
<properties>
<name>time_slot</name><value>16-18</value></properties>
<properties>
<name>service_window_start</name><value>16:00:00</value></properties>
<properties>
<name>service_window_end</name><value>18:00:00</value></properties>
<properties>
<name>appt_number</name><value>test_delay_appointment-FDCRPQISF</value></properties>
<properties>
<name>language</name><value>en</value></properties>
<properties>
<name>duration</name><value>78</value></properties>
<properties>
<name>start_time</name><value>2014-08-14 16:00:00</value></properties>
<properties>
<name>status</name><value>started</value></properties>
<properties>
<name>id</name><value>3998006</value></properties>
<properties>
<name>end_time</name><value>2014-08-15 17:18:00</value></properties>
<properties>
<name>delivery_window_start</name><value>15:30:00</value>
'suspend_activity' and 'prework_activity' Methods

The 'suspend_activity' method can be used for activities with the 'started' or 'pending' status. For started activities, this method changes the activity status to 'pending', makes it not-ordered and creates a new 'suspended' activity duplicating the original one. Properties are updated for both activities, if specified in the request. The 'suspend_activity' method also sets the 'end_time' of the activity with the 'suspended' status to the time when the activity was suspended.

For pending activities, the 'suspend_activity' method makes the activity not-ordered. No new activities are created.

'prework_activity' creates a prework that duplicates the properties of the processed activity and sets its status to 'started' and can be used to update activity properties of the created prework.

'suspend_activity' and 'prework_activity' Requests

Requests of the 'suspend_activity' and 'prework_activity' methods specify:

- activity to be processed
- properties to be updated: for 'suspend_activity' – in the pending not-ordered activity and in the new suspended activity, if applicable for 'prework_activity' – in the created prework
- time of the operation

All requests contain the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity to be processed</td>
</tr>
<tr>
<td>time</td>
<td>Yes</td>
<td>dateTime</td>
<td>which is:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>start time for 'prework_activity'</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>end time of suspended activity for 'suspend_activity'</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(in the YYYY-MM-DD HH:MM:SS format)</td>
</tr>
</tbody>
</table>
## Chapter 2

### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>No</td>
<td>date</td>
<td>date of the operation in the YYYY-MM-DD format default value: current date of the resource</td>
</tr>
</tbody>
</table>

- **'suspend_activity' request example**

```xml
  <SOAP-ENV:Body>
    <ns1:suspend_activity>
      <user>
        <now>2014-08-14T16:52:16Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>90543e80174d51a818dd7210a029be1f</auth_string>
      </user>
      <activity_id>3998011</activity_id>
      <date>2014-08-14</date>
      <time>2014-08-14 16:15:00</time>
    </ns1:suspend_activity>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- **'prework_activity' request example**

```xml
  <SOAP-ENV:Body>
    <ns1:prework_activity>
      <user>
        <now>2014-08-14T16:51:41Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>d699902fadb6b6f6287eaa176e87d3</auth_string>
      </user>
      <activity_id>3998013</activity_id>
      <date>2014-08-14</date>
      <time>2014-08-14 16:00:00</time>
    </ns1:prework_activity>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 'suspend_activity' and 'prework_activity' Responses

The responses of ‘suspend_activity’ and ‘prework_activity’ methods contain data on the method success/failure and all available activity properties.

The responses contain the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>activity</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of activity properties of the processed (new) activity</td>
</tr>
</tbody>
</table>

All available activity properties are returned.
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>

- 'suspend_activity' response example

```xml
  <SOAP-ENV:Body>
    <ns1:suspend_activity_response>
      <result_code>0</result_code>
      <activity>
        <name>name</name>
        <value>Mister Roboto</value>
      </properties>
      <properties>
        <name>zip</name>
        <value>12345</value>
      </properties>
      <properties>
        <name>customer_number</name>
        <value>VGELKFCSMO</value>
      </properties>
      <properties>
        <name>time_zone</name>
        <value>Eastern</value>
      </properties>
      <properties>
        <name>type</name>
        <value>reopened</value>
      </properties>
      <properties>
        <name>position_in_route</name>
        <value>1</value>
      </properties>
      <properties>
        <name>aworktype</name>
        <value>33</value>
      </properties>
      <properties>
        <name>time_slot</name>
        <value>16-18</value>
      </properties>
      <properties>
        <name>service_window_start</name>
        <value>16:00:00</value>
      </properties>
      <properties>
        <name>service_window_end</name>
        <value>18:00:00</value>
      </properties>
      <properties>
        <name>appt_number</name>
        <value>test_suspend_appointment-LSIGBIGCKQ</value>
      </properties>
      <properties>
        <name>language</name>
        <value>en</value>
      </properties>
      <properties>
        <name>start_time</name>
        <value>2014-08-14 16:00:00</value>
      </properties>
      <properties>
        <name>status</name>
        <value>suspended</value>
      </properties>
    </ns1:suspend_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
<properties>
  <name>id</name>
  <value>3998012</value>
</properties>

<properties>
  <name>end_time</name>
  <value>2014-08-14 16:15:00</value>
</properties>

<properties>
  <name>delivery_window_start</name>
  <value>15:30:00</value>
</properties>

<properties>
  <name>delivery_window_end</name>
  <value>16:30:00</value>
</properties>

<properties>
  <name>traveling_time</name>
  <value>30</value>
</properties>

<properties>
  <name>date</name>
  <value>2014-08-14</value>
</properties>

<properties>
  <name>resource_id</name>
  <value>44030</value>
</properties>

</activity>
</ns1:suspend_activity_response>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

- 'prework_activity' response example

```xml
  <SOAP-ENV:Body>
    <ns1:prework_activity_response>
      <result_code>0</result_code>
      <activity>
        <properties>
          <name>name</name>
          <value>Mister Roboto</value>
        </properties>
        <properties>
          <name>zip</name>
          <value>12345</value>
        </properties>
        <properties>
          <name>customer_number</name>
          <value>GRX8W1Q2Q3Q2</value>
        </properties>
        <properties>
          <name>time_zone</name>
          <value>Eastern</value>
        </properties>
        <properties>
          <name>type</name>
          <value>prework</value>
        </properties>
        <properties>
          <name>position_in_route</name>
          <value>1</value>
        </properties>
        <properties>
          <name>aworktype</name>
          <value>81</value>
        </properties>
      </activity>
    </ns1:prework_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
The 'set_required_inventories' method is used to set required inventories for the activity. If any required inventories already exist for the specified activity, the 'set_required_inventories' method updates them according to the request.

'set_required_inventories' Request
The request of 'set_required_inventories' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
</tbody>
</table>
## Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>int</td>
<td>ID of the activity for which required inventory is to be set</td>
</tr>
<tr>
<td>required_inventory</td>
<td>No</td>
<td>struct</td>
<td>element containing the required inventory properties</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If omitted, all required inventories existing for the specified activity are deleted.</td>
</tr>
</tbody>
</table>

- **'required_inventory' Element of 'set_required_inventories' Request**

  The 'required_inventory' element contains the properties of the inventory defined as required for the processed activity.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Yes</td>
<td>string</td>
<td>label of inventory type</td>
</tr>
<tr>
<td>model</td>
<td>No</td>
<td>string</td>
<td>model of inventory</td>
</tr>
<tr>
<td>quantity</td>
<td>Yes</td>
<td>int</td>
<td>how much inventory is required</td>
</tr>
</tbody>
</table>

- **'set_required_inventories' Request Example**

        <SOAP-ENV:Body>  
            <ns1:set_required_inventories>  
                <user>  
                    <now>2014-08-09T09:40:26Z</now>  
                    <login>soap</login>  
                    <company>in132</company>  
                    <auth_string>b3e2f36fa99c4a523c6018eb7b479f44</auth_string>  
                </user>  
                <activity_id>3998006</activity_id>  
                <required_inventory>  
                    <type>NS</type>  
                    <model>XXX</model>  
                    <quantity>2</quantity>  
                </required_inventory>  
                <required_inventory>  
                    <type>NST</type>  
                    <quantity>1</quantity>  
                </required_inventory>  
            </ns1:set_required_inventories>  
        </SOAP-ENV:Body>  
    </SOAP-ENV:Envelope>

- **'set_required_inventories' Response**

  The response of 'set_required_inventories' contains data on the method success/failure and consists of the following elements:
Integrating with Activity Management API

Chapter 2

Activity Management API Overview

### Name | Type | Description
--- | --- | ---
result_code | int | transaction result code
error_msg | string | transaction error description (if ‘result_code’ is other than ‘0’)

- **'set_required_inventories' response Example**

  ```xml
    <SOAP-ENV:Body>
      <ns1:set_required_inventories_response>
        <result_code>0</result_code>
      </ns1:set_required_inventories_response>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
  ```

- **'get_required_inventories' Method**

  The 'get_required_inventories' method is used to retrieve current required inventories of the specified activity.

- **'get_required_inventories' Request**

  The request of 'get_required_inventories' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>int</td>
<td>ID of the activity for which required inventory is to be retrieved</td>
</tr>
</tbody>
</table>

- **'get_required_inventories' Request Example**

  ```xml
    <SOAP-ENV:Body>
      <ns1:get_required_inventories>
        <user>
          <now>2014-08-09T09:40:26Z</now>
          <login>soap</login>
          <company>in132</company>
          <auth_string>b3e2f36fa99c4a523c6018eb7b479f44</auth_string>
        </user>
        <activity_id>3998006</activity_id>
      </ns1:get_required_inventories>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
  ```

- **'get_required_inventories' Response**

  The response of 'set_required_inventories' contains the data on the method success/failure and the list of the required inventories for the specified activity:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code</td>
</tr>
</tbody>
</table>
### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>activity_list</td>
<td>struct</td>
<td>'activity_list' element</td>
</tr>
</tbody>
</table>

- **required_inventory' Element of 'get_required_inventories' Response**

The 'required_inventory' element contains the properties of the inventory defined as required for the processed activity.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>string</td>
<td>label of inventory type</td>
</tr>
<tr>
<td>model</td>
<td>string</td>
<td>model of inventory</td>
</tr>
<tr>
<td>quantity</td>
<td>int</td>
<td>how much inventory is required</td>
</tr>
</tbody>
</table>

- **'get_required_inventories' response Example**

```xml
  <SOAP-ENV:Body>
    <ns1:get_required_inventories_response>
      <result_code>0</result_code>
      <required_inventory>
        <type>NST</type>
        <model/>
        <quantity>1</quantity>
      </required_inventory>
      <required_inventory>
        <type>NS</type>
        <model>XXX</model>
        <quantity>2</quantity>
      </required_inventory>
    </ns1:get_required_inventories_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### Activity Links-Related Methods

#### 'get_activity_links' Method

The 'get_activity_links' method is used to retrieve information about links of an activity.

#### 'get_activity_links' Request

The 'get_activity_links' method request specifies:

- activity for which link data is to be retrieved

The request of 'get_activity_links' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
</tbody>
</table>
### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity for which link data is to be retrieved</td>
</tr>
</tbody>
</table>

- **'get_activity_links' Request Example**

```xml
  <SOAP-ENV:Body>
    <ns1:get_activity_links>
      <user>
        <now>2014-08-14T16:50:47Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>74b18294eb328556832d6d328529bb83</auth_string>
      </user>
      <activity_id>3998006</activity_id>
    </ns1:get_activity_links>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- **'get_activity_links' Response**

The response of 'get_activity_links' contains data on the method success/failure and data on all links of the activity processed.

The response of 'get_activity_links' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>link</td>
<td>struct</td>
<td>details of the found link</td>
</tr>
</tbody>
</table>

- **'link' Element of 'get_activity_links' response**

The 'link' element contains the properties of the activity link found for the specified activity.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_type</td>
<td>enum</td>
<td>label of the link type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: for the link types used in the previous versions ('start-before', 'start-after', 'start-together') the returned link type labels will match the corresponding link type names</td>
</tr>
<tr>
<td>activity_id</td>
<td>int</td>
<td>ID of the activity linked to the processed activity</td>
</tr>
<tr>
<td>resource_id</td>
<td>int</td>
<td>ID of the resource the linked activity is assigned to</td>
</tr>
<tr>
<td>resource_name</td>
<td>string</td>
<td>name of the resource the linked activity is assigned to</td>
</tr>
</tbody>
</table>
## Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>activity_status</td>
<td>string</td>
<td>status of the linked activity</td>
</tr>
<tr>
<td>activity_identifier</td>
<td>string</td>
<td>identification string of the linked activity</td>
</tr>
<tr>
<td>activity_start_time</td>
<td>DateTime</td>
<td>estimated time of arrival of the linked activity (or the actual start_time if the linked activity has been started)</td>
</tr>
<tr>
<td>alerts</td>
<td>int</td>
<td>bit-mask of alert flags applicable for the link (when no alerts exist for the activity link, '0' is returned)</td>
</tr>
<tr>
<td>broken</td>
<td>bool</td>
<td>'false' if link is valid and 'true' otherwise</td>
</tr>
<tr>
<td>min_interval</td>
<td>int</td>
<td>minimal interval between two linked activities</td>
</tr>
<tr>
<td>max_interval</td>
<td>int</td>
<td>maximal interval between two linked activities</td>
</tr>
</tbody>
</table>

- **'alert' Flags and 'alerts' Bit-Mask**

The 'alerts' element is a bit-mask where individual bits are flags meaning certain details about the link:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>'A successor is scheduled prior to the predecessor'</td>
</tr>
<tr>
<td>2</td>
<td>'The predecessor is not completed yet, but a successor is started or has a final status'</td>
</tr>
<tr>
<td>4</td>
<td>'The successor is scheduled prior to a predecessor'</td>
</tr>
<tr>
<td>8</td>
<td>'The successor is pending or started although there is a predecessor that is canceled or not done'</td>
</tr>
<tr>
<td>16</td>
<td>'All predecessors are complete'</td>
</tr>
<tr>
<td>32</td>
<td>'All predecessors have final statuses'</td>
</tr>
<tr>
<td>64</td>
<td>'The activities are not scheduled to the same time inside of the same day or they are scheduled to different days'</td>
</tr>
<tr>
<td>128</td>
<td>'One or more of the related activities is canceled or not done'</td>
</tr>
<tr>
<td>256</td>
<td>'This activity is not started yet, but one or more of the related ones has a final status'</td>
</tr>
<tr>
<td>512</td>
<td>'It is the time to start the activity'</td>
</tr>
</tbody>
</table>
Flag | Meaning
--- | ---
1024 | 'All the related activities are already started'
2048 | 'SW is lost'
4096 | 'Move action: The destination resource has no an appropriate work zone for this activity'
8192 | 'Move action: The destination resource has no an appropriate skill for this activity'
16384 | 'SLA is lost'

• 'get_activity_links' response Example

`<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:toa:activity">`<br>`<SOAP-ENV:Body>`<br>`<ns1:get_activity_links_response>`<br>`<result_code>0</result_code>`<br>`<link>`<br>`<link_type>start-before</link_type>`<br>`<activity_id>3998007</activity_id>`<br>`<resource_id>Mister_NRAWET</resource_id>`<br>`<resource_name>Mister_NRAWET</resource_name>`<br>`<activity_status>pending</activity_status>`<br>`<activity_identifier/>`<br>`<activity_start_time>2014-08-14 10:18:00</activity_start_time>`<br>`<alerts>0</alerts>`<br>`<is_broken>false</is_broken>`<br>`</link>`<br>`</ns1:get_activity_links_response>`<br>`</SOAP-ENV:Body>`<br>`</SOAP-ENV:Envelope>`

'link_activities' and 'unlink_activities' Methods

The 'link_activities' method is used to create a mutual dependency between two specified activities.

The 'unlink_activities' method is used to remove a mutual dependency between two specified activities.

'link_activities' and 'unlink_activities' Request

The request of link_activities' and 'unlink_activities specifies the activities to be linked / unlinked and the type of correlation to be set between them.

Both requests contain the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>from_activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the first linked activity</td>
</tr>
<tr>
<td>to_activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the second linked activity</td>
</tr>
<tr>
<td>link_type</td>
<td>Yes</td>
<td>string</td>
<td>label of the link type</td>
</tr>
</tbody>
</table>
# Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>min_interval</td>
<td>No</td>
<td>int</td>
<td>minimal interval between two linked activities (in minutes) – for the 'link_activities' request only. If omitted, no minimum interval between activities is set.</td>
</tr>
<tr>
<td>max_interval</td>
<td>No</td>
<td>int</td>
<td>maximal interval between two linked activities (in minutes) – for the 'link_activities' request only. If omitted, no maximum interval between activities is set.</td>
</tr>
</tbody>
</table>

- **'link_activities' Request Example**

```xml
  <SOAP-ENV:Body>
    <ns1:link_activities>
      <user>
        <now>2014-08-14T16:50:47Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>74b18294eb328556832d6d328529bb83</auth_string>
      </user>
      <from_activity_id>3998006</from_activity_id>
      <to_activity_id>3998007</to_activity_id>
      <link_type>start-after</link_type>
    </ns1:link_activities>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- **'unlink_activities' Request Example**

```xml
  <SOAP-ENV:Body>
    <ns1:unlink_activities>
      <user>
        <now>2014-08-14T16:50:47Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>74b18294eb328556832d6d328529bb83</auth_string>
      </user>
      <from_activity_id>3998006</from_activity_id>
      <to_activity_id>3998007</to_activity_id>
      <link_type>start-after</link_type>
    </ns1:unlink_activities>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- **'link_activities' and 'unlink_activities' Response**

The response of 'link_activities' and 'unlink_activities' returns data on the method success/failure and consist of the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
</tbody>
</table>
Inventory- Related Methods

'create_customer_inventory' Method
The 'create_customer_inventory' method is used to fill customer pool with inventories.

'create_customer_inventory' Request
The 'create_customer_inventory' method request specifies:

- inventory properties for the inventory to be created in the 'customer' pool
- activity the customer of which owns the inventory

The request of 'create_customer_inventory' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity the customer of which owns the inventory</td>
</tr>
<tr>
<td>properties</td>
<td>No</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of the inventory properties to be set for the created inventory check each property to see if it is mandatory</td>
</tr>
</tbody>
</table>

'create_customer_inventory' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:create_customer_inventory>
      <user>
        <now>2014-08-14T16:49:51Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>7a677f6298c5edaef231ccc454ba4ce0</auth_string>
      </user>
    </ns1:create_customer_inventory>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
The ‘create_customer_inventory’ response returns data and all available properties of the created customer inventory.

The ‘create_customer_inventory’ response consists of the following:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if ‘result_code’ is other than ‘0’)</td>
</tr>
<tr>
<td>inventory</td>
<td>struct</td>
<td>array of ‘properties’ elements each containing one of inventory properties of the created inventory</td>
</tr>
</tbody>
</table>

• ‘create_customer_inventory’ response Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <SOAP-ENV:Body>
    <ns1:create_customer_inventory_response>
      <result_code>0</result_code>
      <inventory>
        <properties>
          <name>serial_number</name>
          <value>test_create_customer_inventory-CCYMJRLHCC</value>
        </properties>
        <properties>
          <name>status</name>
          <value>customer</value>
        </properties>
        <properties>
          <name>type</name>
          <value>Type1</value>
        </properties>
        <properties>
          <name>id</name>
          <value>21034415</value>
        </properties>
        <properties>
          <name>activity_id</name>
          <value>3998006</value>
        </properties>
      </inventory>
    </ns1:create_customer_inventory_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'create_resource_inventory' Method
The 'create_resource_inventory' method is used to fill the resource pool with inventories.

'create_resource_inventory' Request
The 'create_resource_inventory' method request specifies:

- the route for which data is to be retrieved
- properties to be retrieved for the route

The request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>resource_id</td>
<td>Yes</td>
<td>string</td>
<td>external ID of the resource whose route data is to be retrieved</td>
</tr>
<tr>
<td>properties</td>
<td>No</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of the inventory</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>properties to be set for the created inventory</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>check each property to see if it is mandatory</td>
</tr>
</tbody>
</table>

'create_resource_inventory' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:create_resource_inventory>
      <user>
        <now>2014-08-14T16:49:59Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>f0f608813fc9c7d2000c086aaa491456</auth_string>
      </user>
      <resource_id>cable</resource_id>
      <properties>
        <name>type</name>
        <value>Type2</value>
      </properties>
      <properties>
        <name>serial_number</name>
        <value>34534</value>
      </properties>
    </ns1:create_resource_inventory>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'create_resource_inventory' Response
The 'create_resource_inventory' response returns data and all available properties of the created resource inventory.

The 'create_resource_inventory' response consists of the following:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
</tbody>
</table>
### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>inventory</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of inventory properties of the created inventory</td>
</tr>
</tbody>
</table>

- 'create_resource_inventory' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:create_resource_inventory_response>
      <result_code>0</result_code>
      <inventory>
        <properties>
          <name>serial_number</name>
          <value>34534</value>
        </properties>
        <properties>
          <name>type</name>
          <value>Type2</value>
        </properties>
        <properties>
          <name>status</name>
          <value>resource</value>
        </properties>
        <properties>
          <name>id</name>
          <value>21034415</value>
        </properties>
        <properties>
          <name>resource_id</name>
          <value>cable</value>
        </properties>
      </inventory>
    </ns1:create_resource_inventory_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 'get_customer_inventory_list' Method

The 'get_customer_inventory_list' method is used to retrieve data on the inventories in the customer, install and deinstall pools of the specified activity.

#### 'get_customer_inventory_list' Request

The 'get_customer_inventory_list' method request specifies:

- inventory properties to be retrieved for inventories in the 'customer', 'install' and 'deinstall' pool
- activity to which the inventories are assigned

The request of 'get_customer_inventory_list' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
</tbody>
</table>
### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity the customer of which owns the inventory, for which data is to be retrieved</td>
</tr>
<tr>
<td>property_filter</td>
<td>No</td>
<td>struct</td>
<td>each <code>property_filter</code> element contains name of inventory property to be returned in the response. There can be any number of <code>property_filter</code> elements. If omitted, all available properties will be returned.</td>
</tr>
<tr>
<td>select_from</td>
<td>No</td>
<td>int</td>
<td>the number of the inventory in the list starting from which the inventories are to be selected (inventories starting with <code>select_from</code>th are returned). If empty or omitted, all inventories from the first in the list will be returned.</td>
</tr>
<tr>
<td>select_count</td>
<td>No</td>
<td>int</td>
<td>total number of inventories to be returned. If the node is omitted, properties for all inventory in the pool for the specified activity will be returned.</td>
</tr>
</tbody>
</table>

• 'get_customer_inventory_list' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_customer_inventory_list>
      <user>
        <now>2014-08-14T16:50:51Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>0900c5959e9adcb269699659c54c2a08</auth_string>
      </user>
      <activity_id>3998006</activity_id>
    </ns1:get_customer_inventory_list>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

• 'get_customer_inventory_list' Response

The response of `get_customer_inventory_list` returns data on the method success/failure and all available inventory properties specified with `property_filter` that belong to the defined activity.

The response of 'get_customer_inventory_list' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>inventory_list</td>
<td>struct</td>
<td>element containing the list of all found inventories in the pool of the activity and their properties</td>
</tr>
</tbody>
</table>
**'inventory_list' Element of 'get_customer_inventory_list' response**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>int</td>
<td>number of found inventories</td>
</tr>
<tr>
<td>inventories</td>
<td>struct</td>
<td>array of 'inventory' elements each containing properties of one of the inventories returned in the response</td>
</tr>
</tbody>
</table>

**'inventory' Element of 'get_customer_inventory_list' response**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inventory</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of inventory properties of the inventory in the pool for the specified activity</td>
</tr>
</tbody>
</table>

**'get_customer_inventory_list' response Example**

```xml
  <SOAP-ENV:Body>
    <ns1:get_customer_inventory_list_response>
      <result_code>0</result_code>
      <inventory_list>
        <total>2</total>
        <inventories>
          <inventory>
            <properties>
              <name>serial_number</name>
              <value>test_get_customer_inventory_list-MVXGYPFAIH</value>
            </properties>
            <properties>
              <name>status</name>
              <value>customer</value>
            </properties>
            <properties>
              <name>id</name>
              <value>21034415</value>
            </properties>
            <properties>
              <name>activity_id</name>
              <value>3998006</value>
            </properties>
          </inventory>
          <inventory>
            <properties>
              <name>serial_number</name>
              <value>test_get_customer_inventory_list-XNZKEJYDDJ</value>
            </properties>
            <properties>
              <name>status</name>
              <value>customer</value>
            </properties>
            <properties>
              <name>id</name>
              <value>21034416</value>
            </properties>
          </inventory>
        </inventories>
      </inventory_list>
    </ns1:get_customer_inventory_list_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'get_resource_inventory_list' Method

The 'get_resource_inventory_list' method is used to retrieve data on the resource pool inventories of the specified resource.

'get_resource_inventory_list' Request

The 'get_resource_inventory_list' method request specifies:

- inventory properties to be retrieved for inventories in the 'resource' pool
- resource which owns the inventory

The request of 'get_resource_inventory_list' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>resource_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the resource which owns the inventory data for which data is to be returned</td>
</tr>
</tbody>
</table>
| property_filter | No      | struct | each 'property_filter' element contains name of inventory property to be returned in the response  
 |               |          |                                                                             | there can be any number of 'property_filter' elements |
|               |          |        |                                                                             | if omitted, all available properties will be returned |
| select_from   | No       | int    | the number of the inventory in the list starting from which the inventories are to be selected (inventories starting with 'select_from'th are returned) |
|               |          |        |                                                                             | if empty or omitted, all inventories from the first in the list will be returned |
| select_count  | No       | int    | total number of inventories to be returned                                 |
|               |          |        |                                                                             | if omitted, properties for all inventory in the resource pool will be returned |

'get_resource_inventory_list' Request Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <SOAP-ENV:Body>
    <ns1:get_resource_inventory_list>
      <user>
        <now>2014-08-14T16:51:09Z</now>
        <login>soap</login>
      </user>
    </ns1:get_resource_inventory_list>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
The response of 'get_resource_inventory_list' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>inventory_list</td>
<td>struct</td>
<td>element containing the list of all found inventories in the pool of the activity and their properties</td>
</tr>
</tbody>
</table>

- 'inventory_list' Element of 'get_resource_inventory_list' response

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>int</td>
<td>number of found inventories</td>
</tr>
<tr>
<td>inventories</td>
<td>struct</td>
<td>array of 'inventory' elements each containing properties of one of the inventories returned in the response</td>
</tr>
</tbody>
</table>

- 'inventory' Element of 'get_resource_inventory_list' response

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inventory</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of inventory properties of the inventory in the pool for the specified activity</td>
</tr>
</tbody>
</table>

- 'get_resource_inventory_list' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_resource_inventory_list_response>
      <result_code>0</result_code>
      <inventory_list>
        <total>2</total>
        <inventories>
          <inventory>
```

---

ORACLE

97
The 'get_inventory' method is used to retrieve data on the specified inventory.

'get_inventory' Request
The 'get_inventory' method request specifies:

- ID of the inventory to retrieve data for
- properties to be returned for the inventory

The request of 'get_inventory' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>inventory_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the inventory data is to be retrieved for</td>
</tr>
</tbody>
</table>

'get_inventory' Request Example
Chapter 2
Activity Management API Overview

The response of ‘get_inventory’ returns data on the method success/failure and all available properties of the specified inventory.

The response of ‘get_inventory’ contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if ‘result_code’ is other than ‘0’)</td>
</tr>
<tr>
<td>inventory</td>
<td>struct</td>
<td>array of ‘properties’ elements each containing one of inventory properties of the specified inventory</td>
</tr>
</tbody>
</table>

•  ‘get_inventory’ response Example

  <ns1:get_inventory_response>
    <result_code>0</result_code>
    <inventory>
      <properties>
        <name>serial_number</name>
        <value>test_get_inventory-MMBCIATBLQ</value>
      </properties>
      <properties>
        <name>status</name>
        <value>customer</value>
      </properties>
      <properties>
        <name>id</name>
        <value>21034418</value>
      </properties>
      <properties>
        <name>activity_id</name>
        <value>3998009</value>
      </properties>
      <inventory_id>21034418</inventory_id>
    </inventory>
  </ns1:get_inventory_response>
</soap-env:Envelope>
'install_inventory' Method
The 'install_inventory' method is used to move the specified inventory from 'resource' pool to 'install' pool and update the processed inventory properties.

'install_inventory' Request
The request of 'install_inventory' method specifies:

- inventory to be moved from 'resource' pool to 'install' pool
- properties of the processed inventory to be updated

The request of 'install_inventory' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>inventory_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the inventory to be moved to 'install' pool</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity, to 'install' pool of which the inventory is moved</td>
</tr>
<tr>
<td>properties</td>
<td>No</td>
<td>struct</td>
<td>array of 'properties' elements, each containing one of the Inventory Properties to be updated</td>
</tr>
</tbody>
</table>

Check each property to see if it can be updated. When no properties are specified, no properties will be updated.

'install_inventory' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:install_inventory>
      <user>
        <now>2014-08-14T16:51:37Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>17a7fdf6c718ba0149811beaa32d6564</auth_string>
      </user>
      <inventory_id>21034415</inventory_id>
      <activity_id>3998006</activity_id>
    </ns1:install_inventory>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'install_inventory' Response
The response of 'install_inventory' contains data on the method success/failure and all available properties of the installed inventory.

The response of 'install_inventory' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
</tbody>
</table>
Integrating with Activity Management API

Chapter 2

Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>activity_list</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of inventory properties of the installed inventory</td>
</tr>
</tbody>
</table>

- 'install_inventory' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:install_inventory_response>
      <result_code>0</result_code>
      <inventory>
        <properties>
          <name>serial_number</name>
          <value>test_install_inventory-UFBYOMQPGO</value>
        </properties>
        <properties>
          <name>status</name>
          <value>install</value>
        </properties>
        <properties>
          <name>id</name>
          <value>21034415</value>
        </properties>
        <properties>
          <name>activity_id</name>
          <value>3998006</value>
        </properties>
        <properties>
          <name>resource_id</name>
          <value>Mister_NQQDSC</value>
        </properties>
      </inventory>
    </ns1:install_inventory_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'exchange_inventory' Method

The 'exchange_inventory' method is used to install resource inventory and deinstall customer inventory simultaneously.

'exchange_inventory' Request

The 'exchange_inventory' method request specifies:

- inventory to be moved from 'resource' pool to 'install' pool
- another inventory to be moved from 'customer' pool to 'deinstall' pool
- properties to be updated for the inventories

The request of 'exchange_inventory' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>install_inventory_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the inventory to be moved to 'install' pool</td>
</tr>
</tbody>
</table>
**Activity Management API Overview**

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>install_properties</td>
<td>No</td>
<td>struct</td>
<td>array of 'properties' nodes, each containing one of the inventory properties to be updated for the inventory moved from 'resource' to 'install' pool. Check each property to see if it can be updated. When no properties are specified, no properties will be updated.</td>
</tr>
<tr>
<td>deinstall_inventory_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the inventory to be moved to 'deinstall' pool</td>
</tr>
<tr>
<td>deinstall_properties</td>
<td>No</td>
<td>struct</td>
<td>array of 'properties' nodes, each containing one of the inventory properties to be updated for the inventory moved from the 'customer' to 'deinstall' pool. Check each property to see if it can be updated. When no properties are specified, no properties will be updated.</td>
</tr>
</tbody>
</table>

- **'exchange_inventory' Request Example**

  <SOAP-ENV:Body>  
  <ns1:exchange_inventory>  
    <user>  
    <now>2014-08-14T16:50:38Z</now>  
    <login>soap</login>  
    <company>in132</company>  
    <auth_string>8f1f5233b78268ccd90433fd2a3809df</auth_string>  
    </user>  
    <install_inventory_id>21034415</install_inventory_id>  
    <deinstall_inventory_id>21034416</deinstall_inventory_id>  
  </ns1:exchange_inventory>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>`

- **'exchange_inventory' Response**

  The response of 'exchange_inventory' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>activity_list</td>
<td>struct</td>
<td>array of 'properties' nodes that contain all available inventory properties of the installed inventory</td>
</tr>
<tr>
<td>deinstalled_inventory</td>
<td>struct</td>
<td>array of 'properties' nodes that contain all available inventory properties of the deinstalled inventory</td>
</tr>
</tbody>
</table>

- **'exchange_inventory' response Example**

  <SOAP-ENV:Body>  
  <ns1:exchange_inventory_response>  
  <result_code>0</result_code>  
  </ns1:exchange_inventory_response>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>`
'delete_inventory' Method
The 'delete_inventory' method is used to delete inventory from OFSC.

delete_inventory' Request
The 'delete_inventory' method request specifies the inventory to be deleted and contains the following elements:
### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>inventory_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the inventory to be deleted</td>
</tr>
</tbody>
</table>

- **delete_inventory** Request Example

  ```xml
  <SOAP-ENV:Body>
  <ns1:delete_inventory>
  <user>
  <now>2014-08-14T16:50:29Z</now>
  <login>soap</login>
  <company>in132</company>
  <auth_string>034fbd27a9aa205e8cab1b5bf3d284a8</auth_string>
  </user>
  <inventory_id>21034415</inventory_id>
  </ns1:delete_inventory>
  </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
  ```

- **delete_inventory** Response

  ```xml
  <SOAP-ENV:Body>
  <ns1:delete_inventory_response>
  <result_code>0</result_code>
  </ns1:delete_inventory_response>
  </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
  ```

### Other Inventory-Related Methods

- `undo_install_inventory` method is used to move inventory from 'install' pool to 'resource' pool and update the processed inventory's properties
- `deinstall_inventory` method is used to move inventory from 'customer' pool to 'deinstall' pool and update the processed inventory's properties
- `undo_deinstall_inventory` method is used to move inventory from 'deinstall' pool to 'customer' pool and update the processed inventory's properties

### Other Inventory-Related Requests

The requests of `update_inventory`, `undo_install_inventory`, `deinstall_inventory` and `undo_deinstall_inventory` specify:

- inventory to be processed
• specific inventory properties to be updated

All requests contain the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>inventory_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the inventory to be processed</td>
</tr>
<tr>
<td>properties</td>
<td>No</td>
<td>struct</td>
<td>array of 'properties' nodes, each containing one of the inventory properties to be updated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Check each property to see if it can be updated. When no properties are specified, no properties will be updated.</td>
</tr>
</tbody>
</table>

• 'update_inventory' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:update_inventory>
      <user>
        <now>2014-08-14T16:52:42Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>a359209cb119762540f9d92587ac5841</auth_string>
      </user>
      <inventory_id>21034415</inventory_id>
      <properties>
        <name>serial_number</name>
        <value>XXX</value>
      </properties>
    </ns1:update_inventory>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

• 'undo_install_inventory' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:undo_install_inventory>
      <user>
        <now>2014-08-14T16:52:34Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>5c9c744e6e86a3b178cf559c346ea4da</auth_string>
      </user>
      <inventory_id>21034419</inventory_id>
    </ns1:undo_install_inventory>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

• 'deinstall_inventory' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:deinstall_inventory>
      <user>
        <now>2014-08-14T16:50:13Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>df23e70f6f2b1f0d281921895459109d</auth_string>
      </user>
    </ns1:deinstall_inventory>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
• 'undo_deinstall_inventory' Request Example

```
  <ns1:undo_deinstall_inventory>
    <user>
      <now>2014-08-14T16:52:30Z</now>
      <login>soap</login>
      <company>in132</company>
      <auth_string>6b88bb16d64b7aebcfa88ce1c673d064</auth_string>
    </user>
    <inventory_id>21034421</inventory_id>
  </ns1:undo_deinstall_inventory>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Other Inventory-Related Responses

The responses of all methods contain data on the method success/failure and all available inventory properties of the processed inventory.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>inventory</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of inventory properties of the processed inventory</td>
</tr>
</tbody>
</table>

• 'update_inventory' Response Example

```
  <ns1:update_inventory_response>
    <result_code>0</result_code>
    <inventory>
      <properties>
        <name>serial_number</name>
        <value>XXX</value>
      </properties>
      <properties>
        <name>status</name>
        <value>customer</value>
      </properties>
      <properties>
        <name>id</name>
        <value>21034415</value>
      </properties>
      <properties>
        <name>activity_id</name>
        <value>3998006</value>
      </properties>
    </inventory>
  </ns1:update_inventory_response>
</SOAP-ENV:Body>
```
'undo_install_inventory' Response Example

```xml
  <SOAP-ENV:Body>
    <ns1:undo_install_inventory_response>
      <result_code>0</result_code>
      <inventory>
        <properties>
          <name>serial_number</name>
          <value>test_undo_install_inventory-VMXNPUYWZG</value>
        </properties>
        <properties>
          <name>status</name>
          <value></value>
        </properties>
        <properties>
          <name>id</name>
          <value>21034419</value>
        </properties>
        <properties>
          <name>resource_id</name>
          <value>Mister_IPROYH</value>
        </properties>
      </inventory>
    </ns1:undo_install_inventory_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'deinstall_inventory' Response Example

```xml
  <SOAP-ENV:Body>
    <ns1:deinstall_inventory_response>
      <result_code>0</result_code>
      <inventory>
        <properties>
          <name>serial_number</name>
          <value>test_deinstall_inventory-MZNVQMRYTB</value>
        </properties>
        <properties>
          <name>status</name>
          <value></value>
        </properties>
        <properties>
          <name>id</name>
          <value>21034420</value>
        </properties>
        <properties>
          <name>activity_id</name>
          <value>3998006</value>
        </properties>
        <properties>
          <name>resource_id</name>
          <value>Mister_BAEIWQ</value>
        </properties>
      </inventory>
    </ns1:deinstall_inventory_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'undo_deinstall_inventory' Response Example

```xml
  <SOAP-ENV:Body>
    <ns1:undo_deinstall_inventory_response>
      <result_code>0</result_code>
      <inventory>
        <properties>
          <name>serial_number</name>
          <value>test_undo_deinstall_inventory-MZNVQMRYTB</value>
        </properties>
        <properties>
          <name>status</name>
          <value></value>
        </properties>
        <properties>
          <name>id</name>
          <value>21034420</value>
        </properties>
        <properties>
          <name>activity_id</name>
          <value>3998006</value>
        </properties>
        <properties>
          <name>resource_id</name>
          <value>Mister_BAEIWQ</value>
        </properties>
      </inventory>
    </ns1:undo_deinstall_inventory_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'update_inventory' Request and Response Example

- 'update_inventory' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:update_inventory>
      <user>
        <now>2014-08-14T16:52:42Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>a359209cb119762540f9d92587ac5841</auth_string>
      </user>
      <inventory_id>21034415</inventory_id>
      <properties>
        <name>serial_number</name>
        <value>XXX</value>
      </properties>
    </ns1:update_inventory>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- 'update_inventory' Response Example

```xml
  <SOAP-ENV:Body>
    <ns1:update_inventory_response>
      <result_code>0</result_code>
      <inventory>
        <properties>
          <name>serial_number</name>
          <value>XXX</value>
        </properties>
        <properties>
          <name>status</name>
          <value>customer</value>
        </properties>
        <properties>
          <name>id</name>
          <value>21034421</value>
        </properties>
        <properties>
          <name>activity_id</name>
          <value>3998006</value>
        </properties>
      </inventory>
    </ns1:update_inventory_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'undo_install_inventory' Request and Response Example

- 'undo_install_inventory' Request Example

```xml
  <ns1:undo_install_inventory>
    <user>
      <now>2014-08-14T16:52:34Z</now>
      <login>soap</login>
      <company>in132</company>
      <auth_string>5c9c744ae6e86a3b178cf559c346ea4da</auth_string>
    </user>
    <inventory_id>21034419</inventory_id>
  </ns1:undo_install_inventory>
</SOAP-ENV:Envelope>
```

- 'undo_install_inventory' Response Example

```xml
  <ns1:undo_install_inventory_response>
    <result_code>0</result_code>
    <inventory>
      <properties>
        <name>serial_number</name>
        <value>test_undo_install_inventory-VMXNPUYWZG</value>
      </properties>
      <properties>
        <name>status</name>
        <value>resource</value>
      </properties>
      <properties>
        <name>id</name>
        <value>21034419</value>
      </properties>
      <properties>
        <name>resource_id</name>
        <value>Mister_IPROYH</value>
      </properties>
    </inventory>
  </ns1:undo_install_inventory_response>
</SOAP-ENV:Envelope>
```

'deinstall_inventory' Request and Response Example

- 'deinstall_inventory' Request Example

```xml
  <ns1:deinstall_inventory>
    <user>
      <now>2014-08-14T16:50:13Z</now>
      <login>soap</login>
      <company>in132</company>
      <auth_string>df23e70f6f2b1f0d281921895459109d</auth_string>
    </user>
    <inventory_id>21034420</inventory_id>
  </ns1:deinstall_inventory>
</SOAP-ENV:Envelope>
```

- 'deinstall_inventory' Response Example

```xml
  <ns1:deinstall_inventory_response>
    <inventory>
      <properties>
        <name>serial_number</name>
        <value teste_install_inventory-VMXNPUYWZG</value>
      </properties>
      <properties>
        <name>status</name>
        <value>resource</value>
      </properties>
      <properties>
        <name>id</name>
        <value>21034420</value>
      </properties>
      <properties>
        <name>resource_id</name>
        <value>Mister_IPROYH</value>
      </properties>
    </inventory>
  </ns1:deinstall_inventory_response>
</SOAP-ENV:Envelope>
```
'deinstall_inventory' Response Example

```
  <SOAP-ENV:Body>
    <ns1:deinstall_inventory_response>
      <result_code>0</result_code>
      <inventory>
        <properties>
          <name>serial_number</name>
          <value>test_deinstall_inventory-MZNVQMRYTB</value>
        </properties>
        <properties>
          <name>status</name>
          <value>deinstall</value>
        </properties>
        <properties>
          <name>id</name>
          <value>21034420</value>
        </properties>
        <properties>
          <name>activity_id</name>
          <value>3998006</value>
        </properties>
        <properties>
          <name>resource_id</name>
          <value>Mister_BAELWQ</value>
        </properties>
      </inventory>
    </ns1:deinstall_inventory_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'undo_deinstall_inventory' Request and Response Example

**'undo_deinstall_inventory' Request Example**

```
  <SOAP-ENV:Body>
    <ns1:undo_deinstall_inventory>
      <user>
        <now>2014-08-14T16:52:30Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>6b88bb16d64b7eabcfa88ce1c673d064</auth_string>
      </user>
      <inventory_id>21034421</inventory_id>
    </ns1:undo_deinstall_inventory>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**'undo_deinstall_inventory' Response Example**

```
  <SOAP-ENV:Body>
    <ns1:undo_deinstall_inventory_response>
      <result_code>0</result_code>
      <inventory>
        <properties>
          <name>serial_number</name>
          <value>test_undo_deinstall_inventory-NCYIPCNHWI</value>
        </properties>
        <properties>
          <name>status</name>
          <value>customer</value>
        </properties>
      </inventory>
    </ns1:undo_deinstall_inventory_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Request-Related Methods

'create_resource_request' Method

The 'create_resource_request' method is used to create a service request, assigned to a resource in OFSC.

'create_resource_request' Request

The 'create_resource_request' method request specifies:

- resource for whom the service request is to be created
- properties to be set for the service request created

Request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>date</td>
<td>Yes</td>
<td>string</td>
<td>date of the route for which data is to be retrieved in the YYYY-MM-DD format</td>
</tr>
<tr>
<td>resource_id</td>
<td>Yes</td>
<td>string</td>
<td>external ID of the resource whose route data is to be retrieved</td>
</tr>
<tr>
<td>properties</td>
<td>No</td>
<td>struct</td>
<td>array of 'properties' nodes, each containing one of the service request properties for the service request to be created check each property to see if it is mandatory</td>
</tr>
</tbody>
</table>

'create_resource_request' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:create_resource_request>
      <user>
        <now>2014-08-14T16:50:08Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>bcf34c502a8e89134fccc61f6e1eb94f</auth_string>
      </user>
      <date>2014-08-14</date>
      <resource_id>cable</resource_id>
      <properties>
        <name>subject</name>
        <value>Small</value>
        ...
      </properties>
    </ns1:create_resource_request>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'create_resource_request' Response

The response of 'create_resource_request' contains data and all properties of the created service request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>request</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of request properties of the created request</td>
</tr>
</tbody>
</table>

'create_resource_request' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:create_resource_request_response>
      <result_code>0</result_code>
      <request>
        <properties>
          <name>type</name>
          <value>PM</value>
        </properties>
        <properties>
          <name>created</name>
          <value>2014-08-14 16:50:08</value>
        </properties>
        <properties>
          <name>user_id</name>
          <value>4352</value>
        </properties>
        <properties>
          <name>uname</name>
          <value>soap</value>
        </properties>
        <properties>
          <name>resource_id</name>
          <value>cable</value>
        </properties>
        <properties>
          <name>date</name>
          <value>2014-08-14</value>
        </properties>
        <properties>
          <name>subject</name>
          <value>Small</value>
        </properties>
      </request>
    </ns1:create_resource_request_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'get_resource_request_list' Method

The 'get_resource_request_list' method is used to retrieve data on all service requests assigned to the specified resource in OFSC.

'get_resource_request_list' Request

The 'get_resource_request_list' method request specifies:

- resource for which the service request data is to be retrieved
- date service request properties to be retrieved
- specific service request properties to be retrieved
- number of service requests to return from the list of found service requests

The request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>date</td>
<td>Yes</td>
<td>string</td>
<td>date of the route for which data is to be retrieved in the YYYY-MM-DD format</td>
</tr>
<tr>
<td>resource_id</td>
<td>Yes</td>
<td>string</td>
<td>external ID of the resource whose route data is to be retrieved</td>
</tr>
</tbody>
</table>
| property_filter  | No       | struct | each 'property_filter' element contains name of the activity property to be returned in the response  
|                  |          |        | there can be any number of 'property_filter' elements                       |
|                  |          |        | If empty or omitted, all available properties will be returned.              |
| select_from      | No       | int    | the number of the activity in the route starting from which the activities are to be selected (activities starting with 'select_from'th are returned) |
|                  |          |        | If empty or omitted, all activities in the route and their properties are returned |
| select_count     | No       | int    | total number of activities for which data is to be returned                 |
|                  |          |        | If empty or omitted, properties for all activities in the route starting from 'select_from'th will be returned. |

- 'get_resource_request_list' Request Example

```xml
  <body>
    <Big/>
  </body>
```
Oracle Field Service Cloud
Integrating with Activity Management API

Chapter 2
Activity Management API Overview

  <SOAP-ENV:Body>
    <nsl:get_resource_request_list>
      <user>
        <now>2014-08-14T16:51:18Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>d41017e30f868239de0700557087d56a</auth_string>
      </user>
      <date>2014-08-14</date>
      <resource_id>cable</resource_id>
    </nsl:get_resource_request_list>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

'get_resource_request_list' Response
The response of 'get_resource_request_list' contains data and available request properties specified in the request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>request_list</td>
<td>struct</td>
<td>element containing the list of requests of the specified resource for the specified date</td>
</tr>
<tr>
<td>activity_list</td>
<td>struct</td>
<td>'activity_list' element</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>int</td>
<td>number of found requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>array of 'request' elements each containing properties of one of the requests returned in the response</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>struct</td>
<td>array of 'request' elements each containing properties of one of the requests returned in the response</td>
</tr>
</tbody>
</table>

• 'get_resource_request_list' response Example

  <SOAP-ENV:Body>
    <nsl:get_resource_request_list_response>
      <result_code>0</result_code>
      <request_list>
        <total>1</total>
        <requests>
          <request>
            114
          </request>
        </requests>
    </nsl:get_resource_request_list_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
'create_customer_request' Method
The 'create_customer_request' method is used to create a service request assigned to an activity in OFSC.

'create_customer_request' Request
The 'create_customer_request' method request specifies:

- activity for which the service request is to be created
- properties to be set for the service request created

The request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity for which the service request is created</td>
</tr>
<tr>
<td>Name</td>
<td>Required</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>properties</td>
<td>No</td>
<td>struct</td>
<td>array of ‘properties’ elements, each containing one of the service request for the service request to be created. Check each property to see if it is mandatory.</td>
</tr>
</tbody>
</table>

- **'create_customer_request' Request Example**

```xml
  <SOAP-ENV:Body>
    <ns1:create_customer_request>
      <user>
        <now>2014-08-14T16:49:55Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>ae622c4f2dbe9d1a6237468f39400bec</auth_string>
      </user>
      <activity_id>3998006</activity_id>
      <properties>
        <name>subject</name>
        <value>Small</value>
      </properties>
      <properties>
        <name>body</name>
        <value>Big</value>
      </properties>
      <properties>
        <name>type</name>
        <value>SR</value>
      </properties>
    </ns1:create_customer_request>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- **'create_customer_request' Response**

The response of ‘create_customer_request’ contains data and all properties of the created service request.

```xml
  <SOAP-ENV:Body>
    <ns1:create_customer_request_response>
      <result_code>0</result_code>
      <request>
        <properties>
          <name>type</name>
        </properties>
      </request>
    </ns1:create_customer_request_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- **'create_customer_request' response Example**

```xml
  <SOAP-ENV:Body>
    <ns1:create_customer_request_response>
      <result_code>0</result_code>
      <request>
        <properties>
          <name>type</name>
        </properties>
      </request>
    </ns1:create_customer_request_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'get_customer_request_list' Method
The 'get_customer_request_list' method is used to retrieve data on all service requests, assigned to the specified activity in OFSC.

'get_customer_request_list' Request
The 'get_customer_request_list' method request specifies:

- activity for which the service request data is to be retrieved
- date service request properties to be retrieved for
- specific service request properties to be retrieved
- number of service requests to return from the list of found service requests

The request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
</tbody>
</table>
Chapter 2  
Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>Yes</td>
<td>string</td>
<td>date of the route for which data is to be retrieved in the YYYY-MM-DD format</td>
</tr>
<tr>
<td>activity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the activity for which service request data should be retrieved</td>
</tr>
</tbody>
</table>
| property_filter | No       | struct | each ‘property_filter’ element contains name of the activity property to be returned in the response  
|               |          |        | there can be any number of ‘property_filter’ elements                      |
|               |          |        | If empty or omitted, all available properties will be returned.             |
| select_from   | No       | int    | the number of the activity in the route starting from which the activities are to be selected (activities starting with ‘select_from’th are returned) |
|               |          |        | If empty or omitted, all activities in the route and their properties are returned |
| select_count  | No       | int    | total number of activities for which data is to be returned                |
|               |          |        | If empty or omitted, properties for all activities in the route starting from ‘select_from’th will be returned |

• ‘get_customer_request_list’ Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_customer_request_list>
      <user>
        <now>2014-08-21T12:18:34Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>d8267d522522ed49d99950f84754f83c</auth_string>
      </user>
      <activity_id>3998006</activity_id>
    </ns1:get_customer_request_list>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

‘get_customer_request_list’ Response

The response of ‘get_customer_request_list’ contains data and available request properties specified in the request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if ‘result_code’ is other than ‘0’)</td>
</tr>
</tbody>
</table>
### Activity Management API Overview

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>request_list</td>
<td>struct</td>
<td>element containing the list of requests of the specified activity for the specified date</td>
</tr>
<tr>
<td>total</td>
<td>int</td>
<td>total amount of activities in the processed route</td>
</tr>
<tr>
<td>requests</td>
<td>struct</td>
<td>array of 'request' elements each containing properties of one of the requests returned in the response</td>
</tr>
<tr>
<td>request</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of request properties of the customer request</td>
</tr>
</tbody>
</table>

- 'get_customer_request_list' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_customer_request_list_response>
      <result_code>0</result_code>
      <request_list>
        <total>1</total>
        <requests>
          <request>
            <properties>
              <name>type</name> <value>SR</value>
              <name>created</name> <value>2014-08-21 12:18:34</value>
              <name>activity_id</name> <value>3998006</value>
              <name>user_id</name> <value>4352</value>
              <name>uname</name> <value>soap</value>
              <name>id</name> <value>2</value>
              <name>resource_id</name> <value>44030</value>
            </properties>
          </request>
        </requests>
      </request_list>
    </ns1:get_customer_request_list_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'create_inventory_request' Method
The 'create_inventory_request' method is used to create a service request assigned to a specific inventory in OFSC.

'create_inventory_request' Request
The 'create_inventory_request' method request specifies:

- inventory for which the service request is to be created
- properties to be set for the service request created

The request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>inventory_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the inventory for which service request is created</td>
</tr>
<tr>
<td>properties</td>
<td>No</td>
<td>struct</td>
<td>array of 'properties' elements, each containing one of the service request for the service request to be created check each property to see if it is mandatory</td>
</tr>
</tbody>
</table>

'create_inventory_request' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:create_inventory_request>
      <user>
        <now>2014-08-14T16:49:59Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>f0f608813fc9c7d2000c086aaa491456</auth_string>
      </user>
      <inventory_id>21034415</inventory_id>
      <properties>
        <name>subject</name>
        <value>Small</value>
      </properties>
    </ns1:create_inventory_request>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'create_inventory_request' Response

The response of 'create_inventory_request' contains data and all properties of the created service request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>request</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of request properties of the created request</td>
</tr>
</tbody>
</table>

• 'create_inventory_request' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:create_inventory_request_response>
      <result_code>0</result_code>
      <request>
        <properties>
          <name>type</name>
          <value>EQ</value>
        </properties>
        <properties>
          <name>created</name>
          <value>2014-08-14 16:49:59</value>
        </properties>
        <properties>
          <name>user_id</name>
          <value>4352</value>
        </properties>
        <properties>
          <name>uname</name>
          <value>soap</value>
        </properties>
        <properties>
          <name>id</name>
          <value>3</value>
        </properties>
        <properties>
          <name>resource_id</name>
          <value>cable</value>
        </properties>
        <properties>
          <name>inventory_id</name>
          <value>21034415</value>
        </properties>
      </request>
    </ns1:create_inventory_request_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'get_inventory_request_list' Method

The 'get_inventory_request_list' method is used to retrieve data on all service requests assigned to the specified inventory in OFSC.

'get_inventory_request_list' Request

The request of 'get_inventory_request_list' method specifies:

- inventory for which the service request data is to be retrieved
- date service request properties to be retrieved for
- specific service request properties to be retrieved
- number of service requests to return from the list of found service requests

The request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>‘user’ structure</td>
</tr>
<tr>
<td>date</td>
<td>Yes</td>
<td>string</td>
<td>date for which the service requests data should be retrieved in YYYY-MM-DD format (meaningful for resource inventory)</td>
</tr>
<tr>
<td>inventory_id</td>
<td>Yes</td>
<td>int</td>
<td>ID of the inventory for which service request data should be retrieved</td>
</tr>
<tr>
<td>property_filter</td>
<td>No发明</td>
<td>string</td>
<td>each ‘property_filter’ element contains name of the request property to be returned in the response if omitted, all available properties will be returned</td>
</tr>
<tr>
<td>select_from</td>
<td>No发明</td>
<td>int</td>
<td>the number of the request in the list starting from which the requests are to be selected (requests starting with ‘select_from’th are returned) if empty or omitted, all requests from the first in the list will be returned</td>
</tr>
</tbody>
</table>
**Activity Management API Overview**

### Request Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>select_count</td>
<td>No</td>
<td>int</td>
<td>total number of service requests for which data should be returned for</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>if empty or omitted, data for all requests starting with 'select_from' th will be returned</td>
</tr>
</tbody>
</table>

- **'get_inventory_request_list' Request Example**

```xml
  <SOAP-ENV:Body>
    <ns1:get_inventory_request_list>
      <user>
        <now>2014-08-14T16:51:05Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>9c3431211b3f9db277c83dabfcd14347</auth_string>
      </user>
      <inventory_id>21034415</inventory_id>
      <date>2014-08-14</date>
    </ns1:get_inventory_request_list>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- **'get_inventory_request_list' Response**

The response of 'get_inventory_request_list' contains data and available request properties specified in the request.

The response of 'get_inventory_request_list' contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>request_list</td>
<td>struct</td>
<td>element containing the list of requests of the specified inventory for the specified date</td>
</tr>
</tbody>
</table>

- **'request_list' Element of 'get_inventory_request_list' response**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>int</td>
<td>total amount of activities in the processed route</td>
</tr>
<tr>
<td>requests</td>
<td>struct</td>
<td>array of 'request' elements each containing properties of one of the requests returned in the response</td>
</tr>
</tbody>
</table>

- **'request' Element of 'get_inventory_request_list' response**
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>struct</td>
<td>array of ‘properties’ elements each containing one of request properties of the inventory service request</td>
</tr>
</tbody>
</table>

- 'get_inventory_request_list' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_inventory_request_list_response>
      <result_code>0</result_code>
      <request_list>
        <total>1</total>
        <requests>
          <request>
            <properties>
              <name>type</name>
              <value>EQ</value>
            </properties>
            <properties>
              <name>created</name>
              <value>2014-08-14 16:51:05</value>
            </properties>
            <properties>
              <name>activity_id</name>
              <value>3998006</value>
            </properties>
            <properties>
              <name>user_id</name>
              <value>4352</value>
            </properties>
            <properties>
              <name>uname</name>
              <value>soap</value>
            </properties>
            <properties>
              <name>id</name>
              <value>3</value>
            </properties>
            <properties>
              <name>resource_id</name>
              <value>44030</value>
            </properties>
            <properties>
              <name>inventory_id</name>
              <value>21034415</value>
            </properties>
            <properties>
              <name>date</name>
              <value>2014-08-14</value>
            </properties>
            <properties>
              <name>subject</name>
              <value>Small</value>
            </properties>
            <properties>
              <name>body</name>
              <value>Big</value>
            </properties>
          </request>
        </requests>
      </request_list>
    </ns1:get_inventory_request_list_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'get_request' Method
The 'get_request' method is used to retrieve service request properties for any specified service request.

'get_request' Request
The 'get_request' method request specifies:

- service request for which data is to be retrieved

The request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>request_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of the service request for which data should be retrieved</td>
</tr>
</tbody>
</table>

• 'get_request' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_request>
      <user>
        <now>2014-08-14T16:51:22Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>7c838ec336874f6e3a031956f38c87</auth_string>
      </user>
      <request_id>3</request_id>
    </ns1:get_request>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'get_request' Response
The response of 'get_request' contains data and all properties of the specified service request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>request</td>
<td>struct</td>
<td>array of 'properties' elements each containing one of request properties of the specified service request</td>
</tr>
</tbody>
</table>

• 'get_request' Response Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_request_response>
      <result_code>0</result_code>
      <request>125</request>
    </ns1:get_request_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
File-Related Methods

'set_file' Method
The 'set_file' method is used to define a file property in OFSC.

'set_file' Request
The 'set_file' method request specifies:

- entity to which the file property is assigned
- label of the set property
- properties of the set file
The ‘set_file’ method request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>‘user’ structure</td>
</tr>
<tr>
<td>entity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of OFSC entity to which the file is assigned</td>
</tr>
<tr>
<td>property_id</td>
<td>Yes</td>
<td>string</td>
<td>label of property to be set as file (Manage Properties &amp; label)</td>
</tr>
<tr>
<td>file_name</td>
<td>Yes</td>
<td>string</td>
<td>name of the file with extension</td>
</tr>
<tr>
<td>file_data</td>
<td>Yes</td>
<td>base64 binary</td>
<td>file data in base64 encoding</td>
</tr>
<tr>
<td>file_mime_type</td>
<td>No</td>
<td>string</td>
<td>MIME type of the file</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the field is omitted, the type is auto-detected, if auto-detection fails, ‘application/octet-stream’ is used.</td>
</tr>
</tbody>
</table>

- ‘set_file’ Request Example

The example sets the image (blarg-21034415.png) encoded in base64 encoding, as property ‘inv_file’ for entity 21034415. Please note, that for the sake of convenience, the ‘file_data’ in the example is not an actual full file.

```
  <SOAP-ENV:Body>
    <ns1:set_file>
      <user>
        <now>2014-08-14T16:54:34Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>36ca6463b5c5635d6e5fa5cf5517108c</auth_string>
      </user>
      <entity_id>21034415</entity_id>
      <property_id>inv_file</property_id>
      <file_name>blarg-21034415.png</file_name>
      <file_data>iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf8/9hAAAABGdBTUEAAK/INwWK6QAAABgAAAAAQAABJRU5ErkJggg==</file_data>
      <file_mime_type>image/png</file_mime_type>
    </ns1:set_file>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'set_file' Response

The response of ‘set_file’ contains data on error/success result and consists of the following elements:
**Activity Management API Overview**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
</tbody>
</table>

- 'set_file' response Example

```xml
  <SOAP-ENV:Body>
    <ns1:set_file_response>
      <result_code>0</result_code>
    </ns1:set_file_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**'get_file' Method**
The 'get_file' method is used to retrieve file property details from OFSC.

**'get_file' Request**
The 'get_file' method request specifies:
- entity to which the file property is assigned
- label of the property to be retrieved

The 'get_file' method request contains the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>entity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of OFSC entity to which the processed file property is assigned</td>
</tr>
<tr>
<td>property_id</td>
<td>Yes</td>
<td>string</td>
<td>label of file property for which data is to be retrieved</td>
</tr>
</tbody>
</table>

- 'get_file' Request Example

```xml
  <SOAP-ENV:Body>
    <ns1:get_file>
      <user>
        <name>2014-08-14T16:54:35Z</name>
        <login>soap</login>
        <company>in132</company>
        <auth_string>4f60b5732e8a6b7092c2f04f34f6cc51</auth_string>
      </user>
      <entity_id>21034415</entity_id>
      <property_id>inv_file</property_id>
    </ns1:get_file>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'get_file' Response

The response of 'get_file' contains data on the method success/failure and 'file_name', 'file_data' and 'file_mime_type' of the file defined in the request.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
<tr>
<td>file_name</td>
<td>string</td>
<td>name of the file with extension</td>
</tr>
<tr>
<td>file_data</td>
<td>base64</td>
<td>file data in base64 encoding</td>
</tr>
<tr>
<td>file_mime_type</td>
<td>string</td>
<td>MIME type of the file</td>
</tr>
</tbody>
</table>

'get_file' response Example

Please note, that for the sake of convenience, the 'file_data' in the example is not an actual full file.

```xml
  <SOAP-ENV:Body>
    <ns1:get_file_response>
      <result_code>0</result_code>
      <file_name>blarg-21034415.png</file_name>
      <file_data>iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf8/9hAAAABGdBTUEAAK/INwWK6QAAABhJREFUeNpiJPQaqHh59GRqY0m2c6QJra7Z4f4KSQjBJVZIhxRixSaShb
Dy95dNwwiOnomiIgQDae+VnWvsehWUEjY/NtVzqQn1R4ufXeZfsm2WgUQgBInA4hhemZgbpcWN9/
XN27PBp1bBdRqJhPqapZ2DOv5+iOwJnweTmTsdjZKi5Ej/uzd+wT1G6AKYgWYyDjJWYFghmzFsbtcY2gsTJwv09/
V7cR7QgAESqSAKwWM9wu/z7a8B7ваhD3Fk+x+kFgsP0a+x+vrfvFneULj/NT4zWgCBYYtjogSghKl465fYvW
+VAtP07IFZrFycOuDE8emPml0FKHYrBkxAmhSMp1ZVVA27GQO2Ad2ap4zg92MDXJ3czFmdf1x05VEcA2MGIC1ZASdesS2cU/dcm4sTBAIzXZmcNakiCb3/
HLRns4Fo2qYhx3WqDxEuLcOgyNan3D14Hif82dbOiYiBGstSjg4majEpl8rpCNUQjJgki0M5GVA1EBFufWf18Bv12b/
Hig6SmAliZdZhcsE6eP7IixAchAtwVXeclMnhprN/+Ih0txFzrrPZvD9RDEEEzHT6LWpTb7q
+HLSDlOmo2oluqYot37BthHdKX0k6pqhO24Dzd9z/6UYgwPSBvV7atFglarFtu5KLpXeX/xs7aR6AAAAAElFTkSuQmCC</file_data>
    </ns1:get_file_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'delete_file' Method

The 'delete_file' method is used to delete file property from OFSC.

'delete_file' Request

The 'delete_file' method request specifies:

- entity to which the file property is assigned
- label of the property to be deleted

The request contains the following elements:
**Activity Management API Overview**

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Yes</td>
<td>struct</td>
<td>'user' structure</td>
</tr>
<tr>
<td>entity_id</td>
<td>Yes</td>
<td>string</td>
<td>ID of OFSC entity to which the processed file property is assigned</td>
</tr>
<tr>
<td>property_id</td>
<td>Yes</td>
<td>string</td>
<td>label of file property data is to be deleted for</td>
</tr>
</tbody>
</table>

- **'delete_file' Request Example**

  ```xml
    <SOAP-ENV:Body>
      <ns1:delete_file>
        <user>
          <now>2014-08-14T16:54:35Z</now>
          <login>soap</login>
          <company>in132</company>
          <auth_string>4f60b5732e8a6b7092c2f04f34f6cc51</auth_string>
        </user>
        <entity_id>21034415</entity_id>
        <property_id>inv_file</property_id>
      </ns1:delete_file>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
  ```

- **'delete_file' Response**

  The response of 'delete_file' contains data on the method success/failure and consists of the following elements:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>int</td>
<td>transaction result_code</td>
</tr>
<tr>
<td>error_msg</td>
<td>string</td>
<td>transaction error description (if 'result_code' is other than '0')</td>
</tr>
</tbody>
</table>

- **'delete_file' response Example**

  ```xml
    <SOAP-ENV:Body>
      <ns1:delete_file_response>
        <result_code>0</result_code>
      </ns1:delete_file_response>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
  ```

**Transaction Errors**

For each request a response is returned. If an error occurs in the course of transaction processing, the corresponding error response is returned. This can be a SOAP fault (wrong or unknown request is sent) or an error response (a valid request contains invalid element(s)).
SOAP Faults
In case of errors standard SOAP Faults are returned. Soap Fault contains the following fields:

<table>
<thead>
<tr>
<th>Soap Fault field</th>
<th>Possible values of this field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>faultcode</td>
<td>Client, Server</td>
<td>This field is always returned.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Client – means that the problem is with the request – either request has incorrect format, or invalid authentication info is supplied etc.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Server – means that the problem is on Oracle Field Service Cloud side.</td>
</tr>
<tr>
<td>faultstring</td>
<td>Authentication Failed, Unknown location, Bad request format, etc</td>
<td>This field is always returned.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It contains human-readable description of error</td>
</tr>
<tr>
<td>faultactor</td>
<td>DISPATCHER, get_capacity, &lt;absent&gt;</td>
<td>This field is optional.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field is for diagnostic purposes and may be ignored by the Client Application.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It signifies which part of Oracle Field Service Cloud system generated the Soap Fault.</td>
</tr>
<tr>
<td>detail</td>
<td>element containing children: errorCode, errorDetail</td>
<td>This field is optional.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This field contains Oracle Field Service Cloud specific subfields: errorCode, errorDetail.</td>
</tr>
</tbody>
</table>

SOAP Fault Example

```xml
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>Bad Request</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Error Responses
All responses contain a result code and description, when applicable (when the ‘result_code’ is other than ‘0’).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result_code</td>
<td>result of the performed operation</td>
</tr>
<tr>
<td></td>
<td>‘result_code’ is returned in every response.</td>
</tr>
<tr>
<td></td>
<td>For a successful transaction ‘result_code’ = 0 is returned.</td>
</tr>
<tr>
<td></td>
<td>If transaction fails, the ‘result_code’ &gt; 0.</td>
</tr>
</tbody>
</table>
Name | Description
--- | ---
error_msg | more specific description of the error

‘error_msg’ is returned only if ‘result_code’ is other than 0.

- Error Response Example

Example of an error response to an Activity Management API request (‘get_file’ method):

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <SOAP-ENV:Body>
    <ns1:get_file_response>
      <result_code>18</result_code>
      <error_msg>Invalid activity_id=2332ee</error_msg>
    </ns1:get_file_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Error Codes

Error codes related to the Activity Management API methods:

<table>
<thead>
<tr>
<th>Code</th>
<th>Error Message Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no error. Request has been successfully processed</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Authentication failed</td>
<td>user authentication was unsuccessful</td>
</tr>
<tr>
<td>4</td>
<td>Permission denied: operation='get_activity', user='admin'</td>
<td>user has no permission for the action</td>
</tr>
<tr>
<td>5</td>
<td>Wrong provider type</td>
<td>action cannot be performed for the resource type</td>
</tr>
<tr>
<td>6</td>
<td>Wrong provider status</td>
<td>action cannot be performed for the resource status</td>
</tr>
<tr>
<td>7</td>
<td>Wrong appointment type</td>
<td>action cannot be performed for the activity type</td>
</tr>
<tr>
<td>8</td>
<td>Activity is not started: id=3998008, status=pending</td>
<td>action cannot be performed for the activity status</td>
</tr>
<tr>
<td>9</td>
<td>Parameter ‘position_in_route’ is equal to ‘activity_id’</td>
<td>wrong activity position within a route</td>
</tr>
<tr>
<td>11</td>
<td>Can’t start activity: 9382912; start_appointment: The appointment starting order is invalid.</td>
<td>activity cannot be started because it is ordered and is not the first pending activity in the route</td>
</tr>
<tr>
<td>12</td>
<td>Ending route error</td>
<td>route is already deactivated or has unfinished activities</td>
</tr>
<tr>
<td>13</td>
<td>Wrong route status</td>
<td>action cannot be performed for the route status</td>
</tr>
<tr>
<td>Code</td>
<td>Error Message Example</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>14</td>
<td>Wrong inventory status</td>
<td>action cannot be performed for the inventory status</td>
</tr>
<tr>
<td>15</td>
<td>Can’t start activity: 9382903: start_appointment: The appointment cannot be started at the specified time.</td>
<td>action cannot be performed at the time (e.g. app. cannot be started/completed at the specified time)</td>
</tr>
<tr>
<td>16</td>
<td>Can’t link activities:9382910: add_appt_link: Action on past date is not allowed.</td>
<td>action cannot be performed on the date (e.g. cancel activity in the past or start in the future)</td>
</tr>
<tr>
<td>17</td>
<td>Can’t create activity: 9382905: insert_appointment: The 'appt.team_id' mandatory field is not assigned.</td>
<td>mandatory field in request is missing in the request</td>
</tr>
<tr>
<td>18</td>
<td>Search failed: 1679041228: search_appointments: ‘gfh’ is not a valid select_count value</td>
<td>wrong value of a parameter in the request</td>
</tr>
<tr>
<td>19</td>
<td>Activity not found: id=7996012</td>
<td>requested object is not found</td>
</tr>
<tr>
<td>20</td>
<td>Can’t update activity: 9382904: update_appointment: Data has been changed</td>
<td>record to be updated changed by another user</td>
</tr>
<tr>
<td>21</td>
<td>Wrong property type</td>
<td>action cannot be performed for the property type</td>
</tr>
<tr>
<td>22</td>
<td>Invalid base64</td>
<td>invalid file encoding</td>
</tr>
<tr>
<td>23</td>
<td>Can’t create activity: 9382903: insert_appointment: Inconsistent data: sla_window_start &gt; sla_window_end (2014-01-26 01:00:00 &gt; 2014-01-22 01:00:00)</td>
<td>invalid request parameters. Request is missing mandatory field, value out of range etc.</td>
</tr>
<tr>
<td>24</td>
<td>Application server is restarting. Re-send request in a minute</td>
<td>application server is restarting. The request needs to be resent in one minute</td>
</tr>
<tr>
<td>25</td>
<td>Wrong inventory type</td>
<td>action cannot be performed for the inventory type</td>
</tr>
<tr>
<td>26</td>
<td>Can’t create inventory: 9382907: insert_inventory: Duplicate inventory: Typeld = 104, Pool = customer, Model = , ProviderId = 0, ApptId = 3998007</td>
<td>inventory cannot be created as such inventory already exists</td>
</tr>
<tr>
<td>27</td>
<td>Cannot set ‘min_interval’ for this kind of link</td>
<td>activity link does not support modification of minimal/maximal interval</td>
</tr>
<tr>
<td>28</td>
<td>Cannot set ‘min_interval’ and ‘max_interval’</td>
<td>invalid interval between linked activities</td>
</tr>
<tr>
<td>Code</td>
<td>Error Message Example</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>29</td>
<td>Can’t link activities:9382904: add_appt_link: Circular appointment links are not allowed: typeID = 2, fromApptID = 3998006, toApptID = 3998007</td>
<td>link to be created is a circular (loop) link</td>
</tr>
<tr>
<td>30</td>
<td>Can’t link activities:9382904: add_appt_link: Duplicate appointment links are not allowed: typeID = 2, fromApptID = 3998006, toApptID = 3998007</td>
<td>link to be created already exists</td>
</tr>
<tr>
<td>31</td>
<td>Visibility error</td>
<td>user has no permission to view the file property</td>
</tr>
<tr>
<td>32</td>
<td>Too many concurrent requests</td>
<td>the maximum number of ‘get_activities’ requests has been exceeded</td>
</tr>
<tr>
<td>33</td>
<td>Request too expensive</td>
<td>the ‘get_activities’ request runs longer than the maximum time</td>
</tr>
<tr>
<td>34</td>
<td>Cannot change status of segmentable activity to X</td>
<td>the status to be set for a segmentable activity is inconsistent with the status of its segments</td>
</tr>
<tr>
<td>100</td>
<td>Internal error</td>
<td>all other cases</td>
</tr>
</tbody>
</table>
OFSC Glossary Keys
Oracle Product Abbreviations Keyword Map