TOA TECHNOLOGIES

Activity Management SDK

for

ETAdirect

Version 4.5

Part Number E62401-01

Table of Content

1 Introduction

1.1 Document Purpose

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

1.2 Scope of the Document

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

The document is up-to-date with ETAdirect version 4.5.15 functionality.

1.3 Target Audience

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

1.4 Glossary

Term	Explanation
Activate Route	Start the work day
Activity	Entity of the ETAdirect system that represents any time-consuming activity of the resource
Activity Status	Dynamic value that corresponds to the state of particular activity execution
API	Application Programming Interface – a particular set of rules and specifications that software programs follow to communicate and interact with each other
Bucket	Entity appearing on the resource tree which can contain resources of a defined type and be assigned activities
Company	1) Legal entity, using ETAdirect 2) Entity that represents a Client in ETAdirect system; company is created by TOA Technologies during the process of implementation
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
ISO 8601 format	See http://en.wikipedia.org/wiki/ISO_8601
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 ETAdirect (to user, resource, activity or inventory). There are fields and custom properties

Term	Explanation
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 "parent-child" 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.
Service Window	Time frame expected by the customer for an activity as scheduled by the company
SOAP	Lightweight protocol for exchange of information in a decentralized, distributed environment
SOAP 1.1	See http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
SOAP Interface	Interface used to receive requests and return responses via SOAP
SOAP Client Application	Application running at the Client's site and providing interaction with ETAdirect server via SOAP
Team	Group of several resources where one or more resources (Team members) assist another resource (team holder)
Teamwork	Feature allowing resources to assist each other in an activity or on an on-going basis
User	1) Person using ETAdirect 2) Entity used for authentication and authorization, allowing people or external software to access ETAdirect
Work Skill	Qualification required to perform an activity

2 Activity Management API Overview

ETAdirect Activity Management Application Programming Interface can be used throughout the activity lifecycle and enables integration of the ETAdirect 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.

2.1 ETAdirect API Versioning

ETAdirect supports two versions of the Activity Management API. Version 1 includes the functionality implemented up to and including ETAdirect version 4.5.14. Version 2 includes the entire functionality of Version 1 as well as new features developed in ETAdirect versions 4.5.15 and later.

Versioning is implemented by appending 'v1' or 'v2' for 'Version 1' and 'Version 2', respectively, to the API URL. When no version number is used in the URL, such URL always leads to Version 1.

Activity Management API URLs:

Version 1	**Version 2**
/soap/activity/v1/ or /soap/activity/	/soap/activity/v2/

Note: this document describes the functionality implemented both in Version 1 and Version 2. Any features available only in Version 2 are marked with **(available only in API Version 2)**.

2.2 ETAdirect Entities Related to the Activity Management API

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

User is the entity that corresponds to a person, group of people or software, accessing ETAdirect 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.

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).

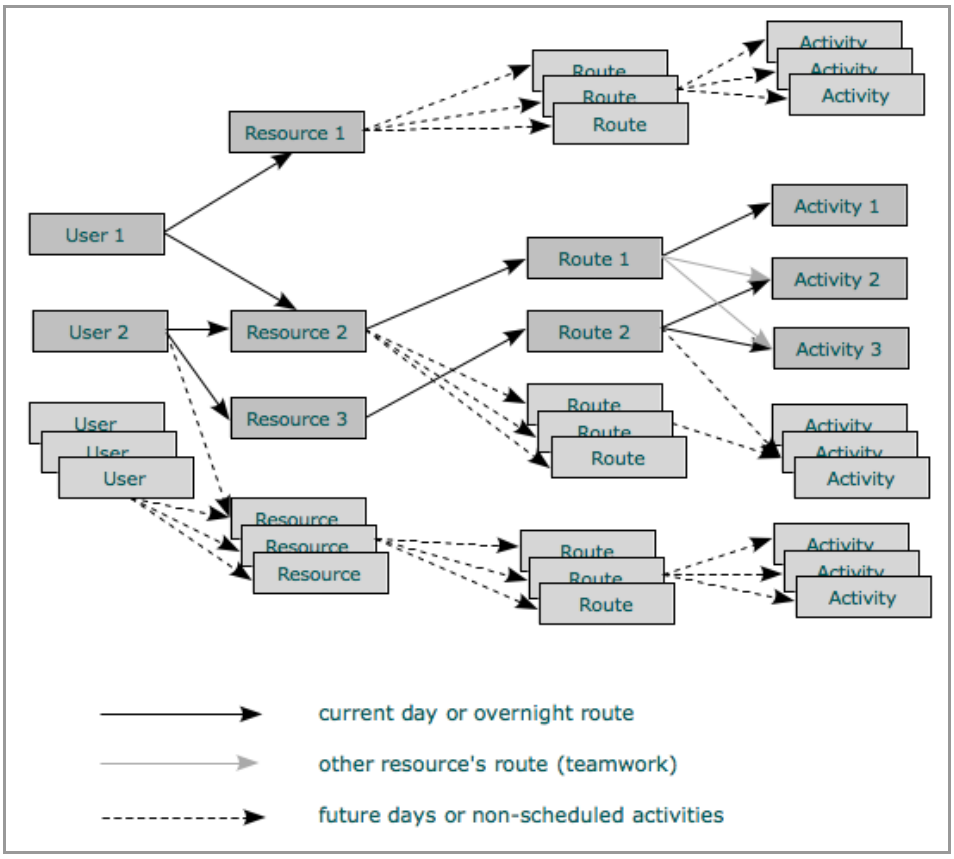

Figure 1: Basic entities correlation

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.

2.3 Entities Not Directly Managed with Activity Management API

2.3.1 User Entity

In the course of ETAdirect implementation for a company, ETAdirect 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 Application → 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'.

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.

2.3.2 Resource Entity

In the course of ETAdirect implementation, ETAdirect 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.

2.3.3 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.

2.3.3.1 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, Activities in the Route, where activity management is described in more details.

2.4 Properties and Fields

Property is a variable associated with some ETAdirect 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 ETAdirect. Some properties are initially

defined in ETAdirect and others are client-specific and created in the course of implementation, though as soon as there is an activity property in ETAdirect, 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 ETAdirect are addressed as 'fields'.

2.4.1 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.

2.4.2 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.

2.4.3 'properties' Structure

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

Name	Required	Type	Description
name	Yes	string	property name, unique for the corresponding property list
value	Yes	string	property value, can be an empty string. When an empty string is sent, the value of the existing property is deleted

2.5 Activities

2.5.1 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 custom 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

2.5.2 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 ETAdirect (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 dedicated section.

2.5.3 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.

2.5.4 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.

2.5.5 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.

2.5.6 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

2.5.7 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.)

2.6 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.

2.6.1 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'.

2.6.2 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.

2.6.3 Activity Statuses

As the resource performs the activity, it changes its status. Using the activity status, ETAdirect 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 ETAdirect, 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.

2.6.3.1 Activity Performance and Status Diagram

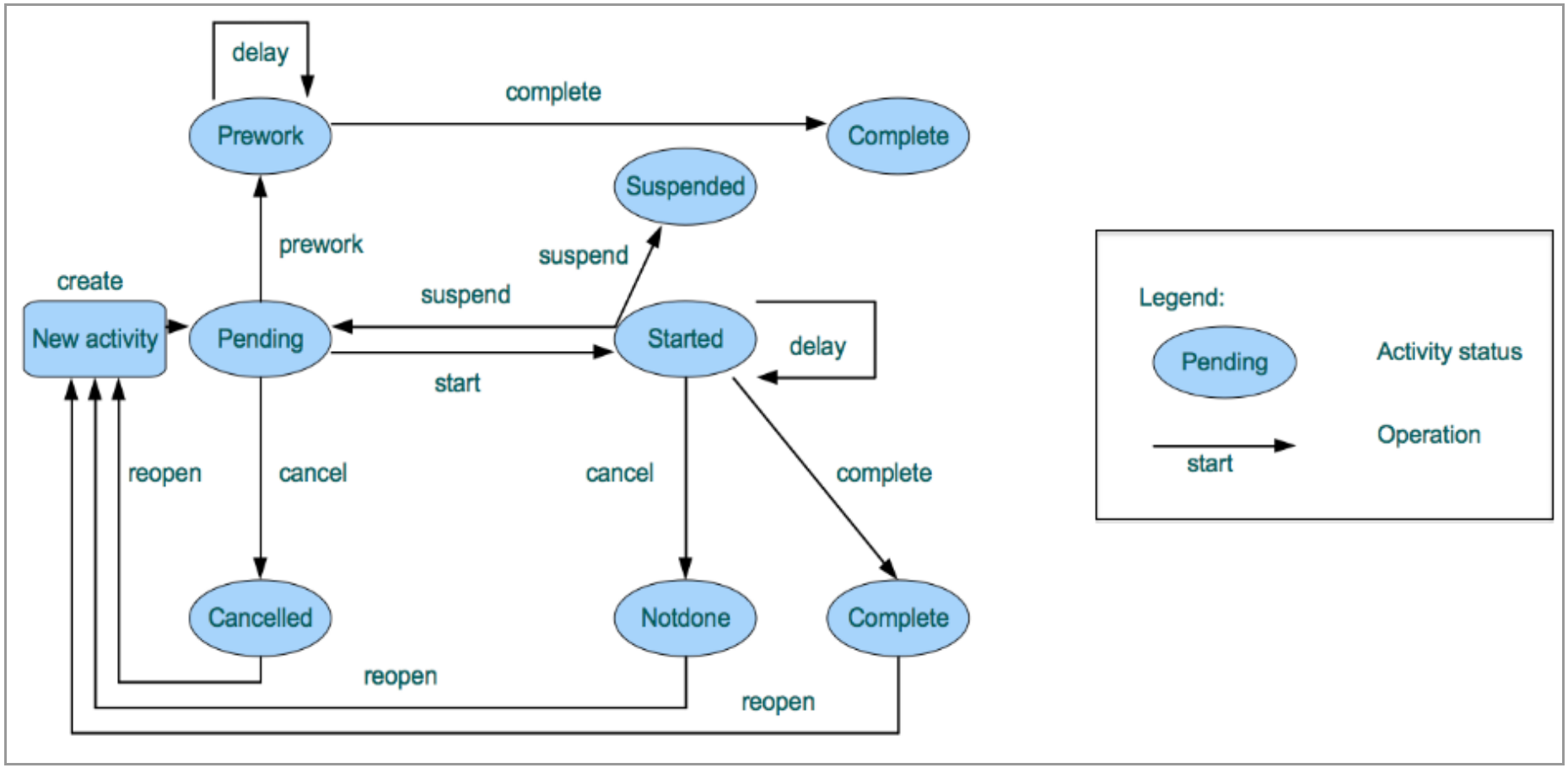

Figure 2: Activity lifecycle

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

2.6.4 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:

2.6.4.1 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'.

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.

2.6.4.2 Preferred Resource Processing

Resources to be handled as required, preferred or forbidden for the activity can be defined using the 'set_resource_preferences' method.

2.6.4.3 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.

2.6.4.4 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.

2.6.4.5 Searching Activities by Multiple Criteria (Available Only in 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.

2.6.4.6 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. Below is the list of activity properties that 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.

2.6.5 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 table below, other activity fields existing in the system and custom activity properties defined in the specific company can also be processed.

Name	Label	Type	Value	Update Allowed
id	aid	int	activity ID	No
appt_number	appt_number	string	work order number	Yes
resource_id	-	string	resource external ID	No
type	atype	enum	activity type **valid values**: regular, prework, reopened	No
status	astatus	enum	activity status **valid values:** pending; started; suspended; canceled; deleted; complete; notdone	No
worktype	aworktype	enum	activity work type (if 'aworktype' is sent, it must contain the activity type ID, if 'worktype' is sent – the activity type label)	No
workzone	aworkzone	enum	activity work zone	No
duration	length	int	length (duration) of the activity in minutes	Not allowed for work types with the 'Define duration manually' feature enabled, otherwise allowed
time_slot	time_slot	enum	activity time slot (string label) available only if time slots are defined for the company (Manage Application → Company Settings → Time Slots)	Yes

Name	Label	Type	Value	Update Allowed
service_window_start	service_window	time	customer service window start time in (H)H:MM format (e.g 8:15, 08:15, and 14:30)	Yes
service_window_end	service_window	time	customer service window end time in (H)H:MM format	Yes
delivery_window_start	delivery_window	time	activity delivery window start in (H)H:MM format (in requests) or HH:MM:SS format (in responses)	Yes
delivery_window_end	delivery_window	time	activity delivery window end in (H)H:MM format (in requests) or HH:MM:SS format (in responses)	Yes
sla_window_start	sla_window_start	DateTime	activity SLA window start in YYYY-MM-DD HH:MM:SS format	Yes
sla_window_end	sla_window_end	DateTime	activity SLA window end in YYYY-MM-DD HH:MM:SS format	Yes
name	cname	string	customer's name	Yes
customer_number	customer_number	string	customer's account number	Yes
phone	cphone	string	customer's regular (land) phone number	Yes
email	cemail	string	customer's email address	Yes
cell	ccell	string	customer's cell phone number	Yes
address	caddress	string	customer's address **Note:** this field is used by geocoding and, therefore, must contain a valid address. Other values will not be resolved correctly by the geocoding server.	Yes
city	ccity	string	customer's city of residence **Note:** this field is used by geocoding and, therefore, must contain a valid city name. Other values will not be resolved correctly by the geocoding server.	Yes

Name	Label	Type	Value	Update Allowed
zip	czip	string	customer's zip/post code **Note:** this field is used by geocoding and, therefore, must contain a valid zip/post code. Other values will not be resolved correctly by the geocoding server.	Yes
state	cstate	string	customer's state of residence **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.	Yes
language	clanguage	enum	notification language company specific language label (en, es, etc.)	Yes
reminder_time	cmessagetime	int	reminder notification time: how many minutes before the activity start time the customer should be notified	Yes
time_zone	c_zid	enum	customer's time zone	Yes
coord_status	acoord_status	string	whether or not activity coordinates were found	No
coordx	acoord_x	float	latitude of the activity (of the customer's location)	Yes
coordy	acoord_y	float	longitude of the activity (of the customer's location)	Yes
start_time	ETA	DateTime	ETA time (for started and ended activities – time when the activity was started) in YYYY-MM-DD HH:MM:SS format	No
end_time	end_time	DateTime	predicted or actual end time of activity in YYYY-MM-DD HH:MM:SS format	No
date	date	date	activity date in YYYY-MM-DD format	No
team_id	-	string	external ID of the team-holder – the head resource within a team	No

Name	Label	Type	Value	Update Allowed
unordered	-	enum	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.)	No
position_in_route	position_in_route	int	if there is a specific time within the resource's route when the activity has to be performed, 'position_in_route' is the number of the activity in the route	Yes

2.6.6 Activity Type Features Affecting Activity Management

Feature	Defines if activities of the type...	If the feature is enabled for an activity...
Teamwork	… are teamwork activities (see the section below for explanation of the term)	… 'team_id' 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
Allow creation in buckets	… can be created in buckets	… 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
Support of not-ordered activities	... can be not-ordered	… not-ordered activity of the type be created if an activity with this feature disabled is defined as not-ordered, the command will fail
Support of non-scheduled activities	… can be activities without a date	… the date of activity can be set to '3000-01-01' to make activity non-scheduled if such date is defined for an activity with the feature disabled, the command will fail
Support of required inventory	… can be assigned required inventory	… required inventory can be assigned to it if required inventory is assigned to an activity with the feature disabled, the command will fail
Support of links	… can be linked to other activities	… it can be linked to other activities if a link is created to an activity with the feature disabled, the command will fail
Support of preferred resources	… can be assigned resource preferences	… resource preferences can be defined for it if resource preference is defined for an activity with the feature disabled, the command will fail

2.6.6.1 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.

2.6.7 Resource Type Features Affecting Activity Management

Feature	Defines if resource of the type...	If the feature is enabled for a resource...
Resource can execute activities	… can have a route, can be assigned activities and can execute them (routes can be started)	… it is possible to process the resource's route, assign activities to it, update and search them and change their statuses
Bucket	… can have a route, can be assigned activities but cannot execute them (routes cannot be started)	… activities can be assigned to the resource's route, updated and searched, but the status of the route and activities cannot be changed (only cancelled)

2.7 Activity Management API Methods

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

Entity	Method	Can be used to...	Details
route	get_route	... retrieve properties of the activities in the specified resource's route for a specified day	Returns the list of activities in the route and their properties visible to the user
	start_route	... start or restart specified resource's route for the current day	Sets the route started
	end_route	... end the specified resource's route for the current day	Sets the route ended
activity	create_activity	... populate the specified route with activities	Creates activities with the specified properties assigned to the specified route
	cancel_activity	... cancel a pending activity before it has been started or cancel a started activity to change it to notdone	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
	prework_activity	... create a prework for activity (applicable for activities with 'pending' status in a started route)	Creates prework in ETAdirect with the 'started' status and sets the prework properties correctly specified in the request and permitted to the user
	start_activity	... set activity status 'started' (applicable for activities with 'pending' status in a started route)	Changes activity status to 'started' and updates all activity properties correctly specified in the request and permitted to the user
	suspend_activity	... change a started activity status to 'pending', make it not-ordered and simultaneously create a suspended activity duplicating the started activity	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
	delay_activity	... prolong the predicted activity duration time (applicable for activities and preworks with the 'started' status)	Changes activity end time and updates all activity properties correctly specified in the request and permitted to the user

Entity	Method	Can be used to...	Details
activity	complete_activity	… set activity status to 'complete' (applicable for activities and preworks with 'started' status)	Changes activity status to 'complete' and updates all activity properties correctly specified in the request and permitted to the user
	reopen_activity	… reopen cancelled, not done or complete activity (applicable for activity with status cancelled, not done or complete not applicable for prework)	Creates a pending activity duplicating the properties of the cancelled, not done or complete activity and in those pending activities updates the properties correctly specified in the request and permitted to the user
	update_activity	… alter activity details (applicable for activities regardless of the status)	Updates all properties correctly specified in the request and permitted to the user
	get_activity	… retrieve specified activity details (applicable for activities regardless of the status)	Returns all properties available in the system and permitted to the user
	get_activities **(available only in API Version 2)**	… search activities by multiple criteria	Returns activities of the specified resource matching the defined search criteria.
	get_activity_work_skills	… retrieve details of specified activity work skills	Returns the list of work skills and their required and preferable qualification level as well as the list of capacity categories if any
	search_activities	… retrieve details of all activities that have the specified value of the specified property	Returns all activities having the same property value available in the system and permitted to the user
activity/ preferences	set_resource_preferences	… define resource preferences	Defines required, preferred and forbidden resources to perform the specific activity
	get_resource_preferences	… retrieve all resource preferences of the specified activity	Returns all preferences defined for the specified activity
activity links	link_activities	… create links	Defines the specified dependency between two specified activities
	unlink_activities	… delete links	Removes the specified dependency between two specified activities
	get_activity_links	… retrieve details of all links of specified activity	Returns details of all links defined for the activity specified
file	set_file	… define a file property	Creates a file property with the defined parameters for the defined entity

Entity	Method	Can be used to...	Details
	get_file	... retrieve details of the specified file property	Returns all available details of the file property
	delete_file	... delete the specified file property	Deletes the file property with the specified label
required inventory	set_required_inventories	... define required inventory for the activity	Adds required inventory to the activity and sets the properties of such inventory
	get_required_inventories	... retrieve required inventory of the activity	Returns the required inventory set for the specified activity together with its properties

2.8 'user' Authentication Structure

All Activity Management API methods use the 'user' structure as request authentication in order to determine the permissions of the ETAdirect client company user. The ETAdirect SOAP authentication structure has the following **mandatory** fields:

Name	Type	Description
now	string	current time in ISO 8601 format
company	string	case-insensitive identifier of the Client for which data is to be retrieved provided by TOA Technologies during integration
login	string	case-insensitive identifier of a specific user within the Company provided by TOA Technologies during integration
auth_string	string	authentication hash; auth_string = md5(now + md5(password)); where 'password' is a case-sensitive set of characters used for user authentication provided by TOA Technologies during integration

For example:

```
<user>
  <now>2014-01-10T13:56:50Z</now>
  <login>soap</login>
  <company>in132</company>
  <auth_string>ba8e6417c60e0e1748265252b4f4d601</auth_string>
</user>
```

2.8.1 Authentication

The node is used for the request authentication. If any of the situations below have place authentication fails and relevant error is returned.

Authentication fails if:

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 ETAdirect
3	login	cannot be found for this company

4 user with this 'login' is not authorized to use the specified method

5 auth_string is not equal to md5(now+md5(password))

For example:
'now' = "2005-07-07T09:25:02+00:00" and password = "Pa$$w0rD"
then
md5 (password) = "06395148c998f3388e87f222bfd5c84b"
concatenated string =
= "2005-0707T09:25:02+00:0006395148c998f3388e87f222bfd5c84b"
auth_string should be:
auth_string = "62469089f554d7a38bacd9be3f29a989"

Otherwise authentication is successful and the request is processed further.

3 Detailed Methods Description

3.1 Route-Related Methods

3.1.1 '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.

3.1.1.1 '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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
date	Yes	sting	date of the route for which data is to be retrieved in the YYYY-MM-DD format
resource_id	Yes	string	external ID of the resource whose route data is to be retrieved
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_route' Request Example**

```
  <SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1: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>
    </ns1:get_route>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.1.1.2 'get_route' Response

The 'get_route' response contains the following elements:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
activity_list	struct	'activity_list' element

3.1.1.3 '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 ETAdirect 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.

The 'activity_list' element contains the following:

Name	Type	Description
total	int	total amount of activities in the processed route
activities	struct	array of 'activity' elements, each being an array of 'properties' elements each containing one activity property
route_start_time	DateTime	time of the route start (YYYY-MM-DD HH:MM:SS)
route_end_time	DateTime	time of the route end (YYYY-MM-DD HH:MM:SS)
route_reactivation_time	DateTime	time of the route reactivation (YYYY-MM-DD HH:MM:(SS)) **(available only in API Version 2)**

- **'get_route' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:get_route_response>
    <result_code>0</result_code>
      <activity_list>
         <total>6</total>
        <activities>
          <activity>
```

```
            <properties>
              <name>id</name>
              <value>3998006</value>
            </properties>
            <properties>
              <name>appt_number</name>
              <value>test_get_route-ZEYVKEVUOE</value>
            </properties>
            <properties>
              <name>time_slot</name>
              <value>all-day</value>
            </properties>
          </activity>
          <activity>
            <properties>
              <name>id</name>
              <value>3998007</value>
            </properties>
            <properties>
              <name>appt_number</name>
              <value>test_get_route-GNLWSZYGTV</value>
            </properties>
            <properties>
              <name>time_slot</name>
              <value>all-day</value>
            </properties>
          </activity>
        </activities>
              <route_start_time>2014-08-14 06:48:00</route_start_time>
              <route_end_time>2014-08-14 18:22:03</route_end_time>
      </activity_list>
     </ns1:get_route_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.1.2 '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: route cannot be ended if there are pending/started activities/prework.

3.1.2.1 '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:

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

3.1.2.2 '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:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')

3.1.2.3 'start_route' Request and Response Example

- **'start_route' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1: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>
    </ns1:start_route>
```

```
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- **'start_route' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:start_route_response>
      <result_code>0</result_code>
    </ns1:start_route_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.1.2.4 'end_route' Request and Response Example

- **'end_route' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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:Envelope>
```

- **'end_route' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:end_route_response>
      <result_code>0</result_code>
    </ns1:end_route_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2 Activity-Related Methods

3.2.1 'create_activity' Method

The 'create_activity' method is used to add a new activity to ETAdirect.

3.2.1.1 '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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
date	Yes	string	date to which activity is to be assigned in the YYYY-MM-DD format
resource_id	Yes	string	external ID of the resource to which the activity is to be assigned
position_in_route	Yes	string	ID of the activity followed by the activity to be created **valid values:** any company-specific activity ID special keys: • **unchanged** – activity position is not changed • **unordered** – created activity is not-ordered • **first** – created activity is the first • **last** – created activity is the last
properties	No	struct	'properties' elements each containing one of activity properties to be set for the new activity there can be as many 'properties' as should be specified for the activity to be created

- **'properties' Array of 'create_activity' Request**

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

Name	Required	Type	Description
worktype	Yes	enum	activity work type (if 'aworktype' is sent, it must contain the activity type ID, if 'worktype' is sent – the activity type label)
duration	Yes	int	length (duration) of the activity in minutes **Note**: this parameter is mandatory for activity types with the 'Define duration manually' feature enabled.
language	Yes	enum	notification language company specific language label (en, es, etc.)
time_zone	Yes	enum	customer's time zone
team_id	Yes	string	external ID of the team-holder – the head resource within a

			team **Note**: this parameter is mandatory for activity types with the 'Teamwork' feature enabled.

- **'create_activity' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

3.2.1.2 'create_activity' Response

The response to 'create_activity' contains data on the method success/failure and all properties of the created activity.

The response of 'create_activity' contains the following elements:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
activity	struct	array of 'properties' elements each containing one of activity properties for new activity All available activity properties are returned.

- **'create_activity' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
    <properties>
      <name>appt_number</name>
      <value>test_cancel_appointment-MOVTQIGRTQ</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>pending</value>
    </properties>
    <properties>
      <name>id</name>
      <value>3998006</value>
    </properties>
    <properties>
      <name>end_time</name>
      <value>2014-08-14 16:48:00</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:create_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.2 'get_activity' Method

The 'get_activity' method is used to retrieve activity properties for the specified activity.

3.2.2.1 'get_activity' Request

The 'get_activity' method request specifies:

- activity to retrieve information for

The request of 'get_activity' contains the following elements:

Name	Required	Type	Description
user	Yes	struct	'user' structure
activity_id	Yes	string	ID of the activity data is to be retrieved for

- **'get_activity' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

3.2.2.2 'get_activity' Response

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

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
activity	struct	array of 'properties' elements each containing one of activity properties for new activity All available activity properties are returned.

- **'get_activity' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
```

```
<ns1:get_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>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>
        <properties>
          <name>start_time</name>
          <value>2014-08-14 16:00:00</value>
        </properties>
        <properties>
          <name>status</name>
          <value>pending</value>
        </properties>
        <properties>
          <name>id</name>
          <value>3998006</value>
        </properties>
        <properties>
          <name>end_time</name>
          <value>2014-08-14 16:48:00</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>
        <properties>
          <name>phone_notification</name>
          <value>1</value>
        </properties>
        <properties>
          <name>email_notification</name>
          <value>1</value>
        </properties>
      </activity>
    </ns1:get_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.3 'get_activities' Method (Available Only in API Version 2)

The method is available only in 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.

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.

3.2.3.1 '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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
resources	Yes	struct	array of 'resource' elements each defining a resource whose activities are to be returned
date_range	Yes	struct	element defining the interval of dates in which activities are to be searched Optionally, the request can be set to include non-scheduled activities.
select_from	No	int	the number of the activity starting from which the activities are to be selected (activities starting with 'select_from'th are returned) **default value**: '1'
select_count	Yes	int	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 less results)
filter_expression	No	struct	element defining the filters which the activities must match to be included in the response If omitted, all activities in the specified date rage are returned.

required_properties	No	struct	array of 'label' elements each containing one of the activity properties to be returned for the found activities **Note**: the 'position_in_route' field cannot be used in the 'get_activities' method. If omitted, only the 'id' field will be returned.

- **'resource' Element of 'get_activities' Request**

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

Name	Required	Type	Description
resource_id	Yes	string	external ID of the resource in whose activities the search is to be performed
include_children	No	enum	defines if activities of subordinate resources are to be returned **valid values:** • **no** – only the specified resource's activities will be returned • **immediate** – activities of the specified resource and the ones directly below it will be returned (first level of descendants) • **all** – activities of the specified resource and of all those below it in the tree hierarchy will be returned **default value:** 'all'

- **'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:

Name	Required	Type	Description
date_from	No	date	beginning of the date interval for the search in the YYYY-MM-DD format
date_to	No	date	end of the date interval for the search in the YYYY-MM-DD format
include_unscheduled	No	bool	flag defining whether unscheduled activities are to be returned in the response **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:

Name	Required	Type	Description
field	Yes	string	field or custom activity property used as the search criterion **Note**: the 'position_in_route' field cannot be used in the 'get_activities' method. For custom properties, property labels are used. **Note**: file properties cannot be used in expressions.
operator	Yes	string	condition used to compare the activity property value with the value specified in the request **valid values:** • **one_of** – the property value is one of the array of values specified in the request • **contains** – the property value contains the specified substring • **has_prefix** – the property value has the specified prefix • **has_suffix** – the property value has the specified suffix • **between** – the property value is in the range between two specified values, both inclusive • **null** – the property specified in the request should be null to match the expression • **equals** – the property value is equal to the specified value • **less_than** – the property value is less than the specified value • **greater_than** – the property value is greater than the specified value • **less_than_or_equal** – the property value is less than or equal to the specified value • **greater_than_or_equal** – the property value is greater than or equal to the specified value **Note**: reverse operators are also possible for each of the above operators. For example: 'null' → 'not_null', 'less_than' → 'not_less_than', 'equals' → 'not_equals', etc.)
value	Yes/No	string	value to which the activity property value is to be compared using the specified operator Mandatory for all operators except 'null'. When the operator is 'null', no 'value' parameter is to be sent.

The 'filter_expression' element defining multiple expressions contains the following parameters:

Name	Required	Type	Description
group_operator	Yes	string	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	Yes	struct	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 example below, 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'.

```
<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>
                  <operator>equals</operator>
                  <value>internal</value>
               </expression>
               <expression>
                  <field>xa_important</field>
                  <operator>equals</operator>
```

```
                    <value>yes</value>
                  </expression>
              </group>
          </filter_expression>
      </group>
</filter_expression>
```

- **'get_activities' Request Example**

The request below 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'.

```
<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>dce625f915d05f1163e40a24b0b068be</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>
               <group_operator>or</group_operator>
```

```
                <group>
                    <filter_expression>
                        <field>traveling_time</field>
                        <operator>equals</operator>
                        <value>15</value>
                    </filter_expression>
                    <filter_expression>
                        <field>name</field>
                        <operator>equals</operator>
                        <value>testrest</value>
                    </filter_expression>
                </group>
            </filter_expression>
        </group>
      </filter_expression>
      <required_properties>
        <label>id</label>
        <label>date</label>
        <label>resource_id</label>
        <label>status</label>
        <label>name</label>
        <label>duration</label>
        <label>traveling_time</label>
      </required_properties>
    </urn:get_activities>
  </soapenv:Body>
</soapenv:Envelope>
```

3.2.3.2 '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:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
expression	struct	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.
activities	struct	array of 'activity' elements each containing properties of one of the activities returned by the search Each 'activity' element is an array of 'properties' elements each containing one of activity properties of the found activity.

		The response contains the activity properties specified in the request.
have_more_results	bool	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.
statistics	struct	element containing the query metrics (for example, execution time, memory usage, etc.) to be used in load estimation and troubleshooting

- **'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:Envelope xmlns:urn="urn:toa:activity"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <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>
            <properties>
               <name>date</name>
               <value>2014-05-21</value>
            </properties>
            <properties>
               <name>resource_id</name>
               <value>Tech 2</value>
            </properties>
            <properties>
               <name>status</name>
               <value>pending</value>
            </properties>
            <properties>
```

```
            <name>name</name>
            <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>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>
      <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>
</soapenv:Envelope>
```

3.2.3.3 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 less 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 less 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.

- **Custom Properties Number Restriction**

The maximum number of custom 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 custom properties used in the 'required_properties' list and in the filter expressions.

3.2.4 '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.

3.2.4.1 '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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
activity_id	Yes	string	ID of the activity for which work skills are to be retrieved

- **'get_activity_work_skills' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

3.2.4.2 'get_activity_work_skills' Response

The 'get_activity_work_skills' response contains data on the method success/failure as well as the following data:

- 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:

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

- **'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:

Name	Type	Description
name	enum	label of the work skill assigned to the activity
description	enum	name of the work skill
required_level	int	required level of the work skill
preferable_level	int	preferable level of the work skill

- **'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:

Name	Type	Description
name	enum	label of the capacity category
description	enum	name of the capacity category

- **'get_activity_work_skills' Response Example**

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
    <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>
```

3.2.5 '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.

3.2.5.1 'update_activity' and 'reopen_activity' Requests

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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
activity_id	Yes	string	ID of the activity to be updated/reopened
position_in_route	Yes	string	ID of the activity followed by the processed activity **valid values:** any company-specific activity ID special keys: • **unchanged** – activity position is not changed • **unordered** – created activity is not-ordered • **first** – created activity is the first • **last** – 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.

3.2.5.2 '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:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
activity	struct	array of 'properties' elements each containing one of activity properties to be set for updated/reopened activity All available activity properties are returned.

3.2.5.3 'update_activity' Request and Response Example

- **'update_activity' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

- **'update_activity' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
        <properties>
          <name>service_window_end</name>
          <value>18:00:00</value>
        </properties>
        <properties>
          <name>appt_number</name>
          <value>test_time_of_booking-HHONMGPGGI</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 17:18:00</value>
        </properties>
        <properties>
          <name>status</name>
          <value>pending</value>
```

```
        </properties>
        <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_start</name>
          <value>16:45:00</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:update_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.5.4 'reopen_activity' Request and Response Example

- **'reopen_activity' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

- **'reopen_activity' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:reopen_activity_response>
      <result_code>0</result_code>
      <activity>
        <properties>
          <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-DGGSDVZRSR</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>
        </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>2014-08-21</value>
        </properties>
        <properties>
          <name>resource_id</name>
          <value>44030</value>
        </properties>
      </activity>
    </ns1:reopen_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.6 'search_activities' Method

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

3.2.6.1 'search_activities' Request

The 'search_activities' 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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
search_in	Yes	string	activity field or property in which the value defined with 'search_for' is to be searched The search is performed in the activity properties specified in ETAdirect (Manage Application → Company Settings → Business Rules → Search Fields). Search Fields can be selected from the available fields (caddress, ccell, cphone, cemail, cname, appt_number, customer_number, czip) and custom properties defined for the activity
search_for	Yes	string	value to search for
date_from	Yes	date	beginning of the date interval for the search in the YYYY-MM-DD format
date_to	Yes	date	end of the date interval for the search in the YYYY-MM-DD format
select_from	Yes	int	the number of the activity starting from which the activities are to be selected (activities starting with 'select_from'th are returned)
select_count	Yes	int	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 less rows)
order	No	enum	enum value defining the order of the activities found **valid values**: • **asc** – ascending • **desc** – descending **default value**: 'desc'
property_filter	No	array	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

- **'search_activities' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:search_activities>
      <user>
        <now>2014-08-14T16:51:53Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>1ff9b3c9461cff5936845c7c65232b74</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>
```

3.2.6.2 'search_activities' Response

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

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
activity_list	struct	list of the found activities with their properties

- **'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.

Name	Type	Description
total	int	number of found activities
activities	array	array of 'activity' elements each containing the properties specified in the 'property_filter' for one activity matching the search criteria if 'property_filter' is empty, 'id', 'resource_id' and 'date' will be returned for all found activities, where available

- **'search_activities' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
            <properties>
              <name>date</name>
              <value>2014-08-14</value>
            </properties>
          </activity>
        </activities>
      </activity_list>
    </ns1:search_activities_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.7 '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.

3.2.7.1 'set_resource_preferences' Request

The 'set_resource_preferences' 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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
activity_id	Yes	string	ID of the activity for which preferences are to be set
preference	No	struct	preference type defined for the activity There can be as many 'preference' elements, as many preferences should be defined for the activity. If omitted, all existing resource preferences for set 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.

Name	Required	Type	Description
resource_id	Yes	string	external ID of the resource for which the preference is set
type	Yes	enum	type of preference: **valid values**: • **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 • **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 • **forbidden** – resources with 'type' set to 'forbidden' cannot be assigned the activity

- **'set_resource_preferences' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

3.2.7.2 'set_resource_preferences' Response

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

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')

- **'set_resource_preferences' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:set_resource_preferences_response>
      <result_code>0</result_code>
    </ns1:set_resource_preferences_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.8 'get_resource_preferences' Method

This method allows retrieving data on the resources preferred, required or forbidden for the activity.

3.2.8.1 'get_resource_preferences' Request

The 'get_resource_preferences' request specifies:

- activity for which the preferences are to be retrieved

The 'set_resource_preferences' request contains the following elements:

Name	Required	Type	Description
user	Yes	struct	'user' structure
activity_id	Yes	string	ID of the activity for which the preferences are to be retrieved

- **'get_resource_preferences' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
</SOAP-ENV:Envelope>
```

3.2.8.2 '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.

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
preference	struct	preference type defined for the user found for the activity there will be as many 'preference' elements, as many preferences are set in the system for the activity

- **'preference' Element of 'get_resource_preferences' Response**

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

Name	Type	Description
resource_id	string	external ID of the resource for which the preference is set
type	enum	type of preference: **valid values**: • **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 • **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 • **forbidden** – resources with 'type' set to 'forbidden' cannot be assigned the activity

- **'get_resource_preferences' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

3.2.9 'delay_activity' Method

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

3.2.9.1 'delay_activity' Request

The 'delay_activity' request specifies:

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

The 'delay_activity' request contains the following elements:

Name	Required	Type	Description
user	Yes	struct	'user' structure
activity_id	Yes	string	ID of the activity to be processed
value	Yes	int	delay in minutes
date	No	date	date of the operation in the YYYY-MM-DD format **default value**: current date for the resource
properties	No	array	array of 'properties' elements each containing one of activity properties to be updated for the processed activity If omitted, no activity properties are updated.

- **'delay_activity' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

3.2.9.2 'delay_activity' Response

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

The response of 'delay_activity' contains the following elements:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
activity	struct	array of 'properties' elements each containing one of activity properties of the processed (updated) activity All available activity properties are returned.

- **'delay_activity' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:delay_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>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-FDCRYPQISF</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>
    </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>act_property</name>
          <value>new property</value>
        </properties>
      </activity>
    </ns1:delay_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.10 Other Activity-Related Methods

The rest of activity-related methods have similar request and response structures.

Most of them, namely: 'start_activity', 'cancel_activity', 'complete_activity' change the activity status in a way defined with their names and can update the specified activity properties.

'suspend_activity' 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.

Note: The 'suspend_activity' request can only be sent for activities with status 'started'. To make a 'pending' activity not-ordered, the 'update_activity' request must be used.

'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 the created prework.

Note: 'cancel_activity' will set status of a pending activity to 'cancelled' and of a started activity to 'notdone'.

3.2.10.1 Other Activity-Related Requests

Requests of 'start_activity', 'cancel_activity', 'complete_activity', 'suspend_activity', 'prework_activity' methods specify:

- activity to be processed
- properties to be updated (for 'start_activity', 'cancel_activity', 'complete_activity' – in the processed activity

  for 'suspend_activity' – in the pending not-ordered activity and in the new suspended activity

  for 'prework_activity' – in the created prework)
- time of the operation

All requests contain the following elements:

Name	Required	Type	Description
user	Yes	struct	'user' structure
activity_id	Yes	string	ID of the activity to be processed
time	Yes	dateTime	which is: start time for 'start_activity' and 'prework_activity' end time for 'cancel_activity', 'complete_activity' end time of suspended activity for 'suspend_activity' (in the YYYY-MM-DD HH:MM:SS format)
date	No	date	date of the operation in the YYYY-MM-DD format **default value**: current date of the resource
properties	No	array	array of 'properties' elements each containing one of activity properties to be updated for the processed activity

3.2.10.2 Other Activity-Related Responses

The responses of 'start_activity', 'cancel_activity', 'complete_activity', 'suspend_activity', 'prework_activity' methods contain data on the method success/failure and all available activity properties.

The responses contain the following elements:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
activity	struct	array of 'properties' elements each containing one of activity properties of the processed (new) activity All available activity properties are returned.

3.2.10.3 'start_activity' Request and Response Example

- **'start_activity' Request**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
</SOAP-ENV:Envelope>
```

- **'start_activity' Response**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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-XWHBQVWRVB</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>
        <properties>
          <name>id</name>
          <value>3998006</value>
        </properties>
        <properties>
          <name>end_time</name>
          <value>2014-08-14 16:48:00</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:start_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Note: as requests and responses of all methods are identical, in the examples below their optional elements are empty.

3.2.10.4 'complete_activity' Request and Response Example

- **'complete_activity' Request**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
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**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:complete_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-XWHBQVWRVB</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>complete</value>
        </properties>
        <properties>
          <name>id</name>
          <value>3998006</value>
        </properties>
        <properties>
          <name>end_time</name>
          <value>2014-08-14 16:10:00</value>
        </properties>
        <properties>
          <name>delivery_window_start</name>
```

```
          <value>15:30:00</value>
        </properties>
        <properties>
          <name>delivery_window_end</name>
          <value>21:00:00</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:complete_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.10.5 'cancel_activity' Request and Response Example

- **'cancel_activity' Request**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:cancel_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>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>
        <properties>
          <name>new_property</name>
          <value>property</value>
        </properties>
      </activity>
    </ns1:cancel_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.10.6 'suspend_activity' Request and Response Example

- **'suspend_activity' Request**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

- **'suspend_activity' Response**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:suspend_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>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>
        <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>
```

3.2.10.7 'prework_activity' Request and Response Example

- **'prework_activity' Request**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:prework_activity>
      <user>
        <now>2014-08-14T16:51:41Z</now>
        <login>soap</login>
        <company>in132</company>
        <auth_string>d699902fdbd666b6f6287eaa176e87d3</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>
```

- **'prework_activity' Response**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>GKXBWKMNQD</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>33</value>
        </properties>
        <properties>
          <name>time_slot</name>
          <value>16-18</value>
        </properties>
        <properties>
          <name>appt_number</name>
          <value>test_prework_appointment-YVOWPRLSXE</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>
        <properties>
          <name>id</name>
          <value>3998014</value>
        </properties>
        <properties>
          <name>end_time</name>
          <value>2014-08-14 16:48:00</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:prework_activity_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.11 'set_required_inventories' Method

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.

3.2.11.1 'set_required_inventories' Request

The request of 'set_required_inventories' contains the following elements:

Name	Required	Type	Description
user	Yes	struct	'user' structure
activity_id	Yes	int	ID of the activity for which required inventory is to be set
required_inventory	No	struct	element containing the required inventory properties If omitted, all required inventories existing for the specified activity are deleted.

- **'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.

Name	Required	Type	Description
type	Yes	string	label of inventory type
model	No	string	model of inventory
quantity	Yes	int	how much inventory is required

- **'set_required_inventories' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

3.2.11.2 'set_required_inventories' Response

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

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**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:set_required_inventories_response>
      <result_code>0</result_code>
    </ns1:set_required_inventories_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.12 'get_required_inventories' Method

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

3.2.12.1 'get_required_inventories' Request

The request of 'get_required_inventories' contains the following elements:

Name	Required	Type	Description
user	Yes	struct	'user' structure
activity_id	Yes	int	ID of the activity for which required inventory is to be retrieved

- **'get_required_inventories' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

3.2.12.2 '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:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
required_inventory	array	array of records containing the required inventory properties

- **'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.

Name	Type	Description
type	string	label of inventory type
model	string	model of inventory
quantity	int	how much inventory is required

- **'get_required_inventories' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

3.3 Activity Links-Related Methods

3.3.1 'get_activity_links' Method

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

3.3.1.1 '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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
activity_id	Yes	string	ID of the activity for which link data is to be retrieved

- **'get_activity_links' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

3.3.1.2 '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:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
link	struct	details of the found link

- **'link' Element of 'get_activity_links' Response**

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

Name	Type	Description
link_type	enum	label of the link type

Name	Type	Description
		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
activity_id	int	ID of the activity linked to the processed activity
resource_id	int	ID of the resource the linked activity is assigned to
resource_name	string	name of the resource the linked activity is assigned to
activity_status	string	status of the linked activity
activity_identifier	string	identification string of the linked activity
activity_start_time	DateTime	estimated time of arrival of the linked activity (or the actual start_time if the linked activity has been started)
alerts	int	bit-mask of alert flags applicable for the link (when no alerts exist for the activity link, '0' is returned)
broken	bool	'false' if link is valid and 'true' otherwise
min_interval	int	minimal interval between two linked activities
max_interval	int	maximal interval between two linked activities

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

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

Flag	Meaning
1	'A successor is scheduled prior to the predecessor'
2	'The predecessor is not completed yet, but a successor is started or has a final status'
4	'The successor is scheduled prior to a predecessor'
8	'The successor is pending or started although there is a predecessor that is canceled or not done'
16	'All predecessors are complete'
32	'All predecessors have final statuses'
64	'The activities are not scheduled to the same time inside of the same day or they are scheduled to different days'
128	'One or more of the related activities is canceled or not done'
256	'This activity is not started yet, but one or more of the related ones has a final status'
512	'It is the time to start the activity'
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">
  <SOAP-ENV:Body>
    <ns1:get_activity_links_response>
      <result_code>0</result_code>
      <link>
        <link_type>start-before</link_type>
        <activity_id>3998007</activity_id>
        <resource_id>Mister_NRAWET</resource_id>
        <resource_name>Mister_NRAWET</resource_name>
        <activity_status>pending</activity_status>
        <activity_identifier/>
        <activity_start_time>2014-08-14 10:18:00</activity_start_time>
        <alerts>0</alerts>
        <is_broken>false</is_broken>
      </link>
    </ns1:get_activity_links_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.3.2 '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.

3.3.2.1 '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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
from_activity_id	Yes	string	ID of the first linked activity
to_activity_id	Yes	string	ID of the second linked activity
link_type	Yes	string	label of the link type
min_interval	No	int	minimal interval between two linked activities (in minutes) – **for the 'link_activities' request only** If omitted, no minimum interval between activities is set.
max_interval	No	int	maximal interval between two linked activities (in minutes) – **for the 'link_activities' request only** If omitted, no maximum interval between activities is set.

3.3.2.2 '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:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')

3.3.2.3 'link_activities' Request and Response Example

- **'link_activities' Request Example**

Note: In the example below, activity 3998007 must start after activity 3998006

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

- **'link_activities' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:link_activities_response>
      <result_code>0</result_code>
    </ns1:link_activities_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.3.2.4 'unlink_activities' Request and Response Example

- **'unlink_activities' Request Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

- **'unlink_activities' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:unlink_activities_response>
      <result_code>0</result_code>
    </ns1:unlink_activities_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.4 File-Related Methods

File-related methods deal with file properties in ETAdirect. File property is a file assigned to a specific activity, resource or inventory in ETAdirect. File data is always encoded using base64 encoding.

3.4.1 'set_file' Method

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

3.4.1.1 '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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
entity_id	Yes	string	ID of ETAdirect entity to which the file is assigned
property_id	Yes	string	label of property to be set as file (Manage Application → Properties → label)
file_name	Yes	string	name of the file with extension
file_data	Yes	base64 binary	file data in base64 encoding
file_mime_type	No	string	MIME type of the file If the field is omitted, the type is auto-detected, if auto-detection fails, 'application/octet-stream' is used.

- **'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:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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/I
NwWK6QAAABl0RVh0U29mdHdhcmUAQWRvYmUgSW1hZ2VSZWFkeXHJZTwAAAIpSURBVDjLddM9aFRBF
Ibh98zM3WyybnYVf4KSQjBJJVZBixhRixSaShtBMKUoWomgnaCxsJdgIQSstE4nEhNREgyoZYhpko
gkuMa4/3fuHIu7gpLd00wz52POMzMydu/Dy958dMwYioomIIgqDa+VnWrzebNUejY/NV6nQ8nlR4u
fXt0fzm2WgxUgqBInAWdhemGbpcWNN9/XN27PPb1QbRdgjEhPqap2ZUv5+iOwvJnweT1mT5djZKjI
6Ej/udz+wt1OJzAKYgWyDjJWyFghmzFsbtcY2gsTJwv09/Vc7RTgAEQgsqAKaoWsM8wu/z7a8B7vA
8cHD3Fr+ktFgspO3a+vrdVfNEulJ/NT4zWngCBYY1oqSghKI465fvYwW+VAatPX07IZmF7YfrC0uD
E8emPmilOFkHYiBKxAxhmSRPlZVVa2FGOU2Ad2ap4zg92MDBXJZczFmdflx05VEcAZMGIIClZASde
sS2cU/dcm4sTBArNzXTcNakiCb3/HLRsn4Fo2qyXh3WqDXzUlcgYnam3Dl4Hif82dbOiyiBGstSjg
4majEpl8rpCNUQUjgkia0M5GVAlBEBFUwflEv12b/Hig6SmA1iDtzhcsE6eP7LIxAchAtwNVxc1Mn
hprN/
+lh0txErxrPZVdFdRDEEzHT6LWpTbtq+HLSDDiOm2o1uqlyOT37bIhHdKaXoL6pqhq24Dzd96/tUY
GwPSBVv7atFglaFIu5KLuPxeX/xsp7aR6AAAAAElFTkSuQmCC</file_data>
      <file_mime_type>image/png</file_mime_type>
    </ns1:set_file>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.4.1.2 'set_file' Response

The response of 'set_file' contains data on error/success result and consists of the following elements:

Name	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')

- **'set_file' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:set_file_response>
      <result_code>0</result_code>
    </ns1:set_file_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.4.2 'get_file' Method

The 'get_file' method is used to retrieve file property details from ETAdirect.

3.4.2.1 '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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
entity_id	Yes	string	ID of ETAdirect entity to which the processed file property is assigned
property_id	Yes	string	label of file property for which data is to be retrieved

- **'get_file' Request Example**

The example is to retrieve 'inv_file' for entity 21034415.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:get_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:get_file>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.4.2.2 '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.

Field	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')
file_name	string	name of the file with extension
file_data	base64 binary	file data in base64 encoding

file_mime_type	string	MIME type of the file

- **'get_file' Response Example**

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

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:get_file_response>
      <result_code>0</result_code>
      <file_name>blarg-21034415.png</file_name>
      <file_data>iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf8/9hAAAABGdBTUEAAK/I
NwWK6QAAABl0RVh0U29mdHdhcmUAQWRvYmUgSW1hZ2VSZWFkeXHJZTwAAAIpSURBVDjLddM9aFRBF
Ibh98zM3WyybnYVf4KSQjBJJVZBixhRixSaShtBMKUoWomgnaCxsJdgIQSstE4nEhNREgyoZYhpko
gkuMa4/3fuHIu7gpLd00wz52POMzMydu/Dy958dMwYioomIIgqDa+VnWrzebNUejY/NV6nQ8nlR4u
fXt0fzm2WgxUgqBInAWdhemGbpcWNN9/XN27PPb1QbRdgjEhPqap2ZUv5+iOwvJnweT1mT5djZKjI
6Ej/udz+wt1OJzAKYgWyDjJWyFghmzFsbtcY2gsTJwv09/Vc7RTgAEQgsqAKaoWsM8wu/z7a8B7vA
8cHD3Fr+ktFgspO3a+vrdVfNEulJ/NT4zWngCBYY1oqSghKI465fvYwW+VAatPX07IZmF7YfrC0uD
E8emPmilOFkHYiBKxAxhmSRPlZVVa2FGOU2Ad2ap4zg92MDBXJZczFmdflx05VEcAZMGIIClZASde
sS2cU/dcm4sTBArNzXTcNakiCb3/HLRsn4Fo2qyXh3WqDXzUlcgYnam3Dl4Hif82dbOiyiBGstSjg
4majEpl8rpCNUQUjgkia0M5GVAlBEBFUwflEv12b/Hig6SmA1iDtzhcsE6eP7LIxAchAtwNVxc1Mn
hprN/
+lh0txErxrPZVdFdRDEEzHT6LWpTbtq+HLSDDiOm2o1uqlyOT37bIhHdKaXoL6pqhq24Dzd96/tUY
GwPSBVv7atFglaFIu5KLuPxeX/xsp7aR6AAAAAElFTkSuQmCC</file_data>
      <file_mime_type>image/png</file_mime_type>
    </ns1:get_file_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.4.3 'delete_file' Method

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

3.4.3.1 '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:

Name	Required	Type	Description
user	Yes	struct	'user' structure
entity_id	Yes	string	ID of ETAdirect entity to which the processed file property is assigned
property_id	Yes	string	label of file property data is to be deleted for

- **'delete_file' Request Example**

The example is to delete property 'inv_file' for entity 21034415.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <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>
```

3.4.3.2 'delete_file' Response

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

Field	Type	Description
result_code	int	transaction result_code
error_msg	string	transaction error description (if 'result_code' is other than '0')

- **'delete_file' Response Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
  <SOAP-ENV:Body>
    <ns1:delete_file_response>
      <result_code>0</result_code>
    </ns1:delete_file_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4 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)).

4.1 SOAP Faults

In case of errors standard SOAP Faults are returned. Soap Fault contains the following fields:

Soap Fault field	Possible values of this field	Description
faultcode	• Client • Server	This field is always returned. • Client – means that the problem is with the request – either request has incorrect format, or invalid authentication info is supplied etc. • Server – means that the problem is on ETAdirect side.
faultstring	• Authentication Failed • Unknown location • Bad request format • etc.	This field is always returned. It contains human-readable description of error
faultactor	• DISPATCHER • API-PORTAL • <absent>	This field is optional. This field is for diagnostic purposes and may be ignored by the Client Application. It signifies which part of ETAdirect system generated the Soap Fault.
detail	free-format element	This field is optional. This field contains ETAdirect specific subfields depending on the module emitting the SOAP Fault. This field is for diagnostic purposes and may be ignored by the Client Application.

- **SOAP Fault Example**

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
     <SOAP-ENV:Fault>
        <faultcode>SOAP-ENV:Client</faultcode>
        <faultstring>Bad Request</faultstring>
     </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.2 Error Responses

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

Field	Description
result_code	result of the performed operation 'result_code' is returned in every response. For a successful transaction 'result_code' = 0 is returned. If transaction fails, the 'result_code' > 0.
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 version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:toa:activity">
 <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>
```

4.3 Error Codes

Error codes related to the Activity Management API methods:

Code	error_msg Example	Description
0		no error. Request has been successfully processed
3	Authentication failed	user authentication was unsuccessful
4	Permission denied: operation='get_activity', user='admin'	user has no permission for the action
8	Activity is not started: id=3998008, status=pending	action cannot be performed for the activity status
9	Parameter 'position_in_route' is equal to 'activity_id'	wrong activity position within a route
11	Can't start activity: 9382912: start_appointment: The appointment starting order is invalid.	activity cannot be started because it is ordered and is not the first pending activity in the route
15	Can't start activity: 9382903: start_appointment: The appointment cannot be started at the specified time.	action cannot be performed at the time (e.g app. cannot be started/complete at the specified time)
16	Can't link activities:9382910: add_appt_link: Action on past date is not allowed.	action cannot be performed on the date (e.g cancel activity in the past or start in the future)

Code	error_msg Example	Description
17	Can't create activity: 9382905: insert_appointment: The 'appt.team_id' mandatory field is not assigned.	mandatory field in request is missing in the request
18	Search failed: 1679041228: search_appointments: 'gfh' is not a valid select_count value	wrong value of a parameter in the request
19	Activity not found: id=7996012	requested object is not found
20	Can't update activity: 9382904: update_appointment: Data has been changed	record to be updated changed by another user
23	Can't create activity: 9382903: insert_appointment: Inconsistent data: sla_window_start > sla_window_end (2014-01-26 01:00:00 > 2014-01-22 01:00:00)	invalid request parameters. Request is missing mandatory field, value out of range etc.
27	Cannot set 'min_interval' for this kind of link	activity link does not support modification of minimal/maximal interval
28	Cannot set 'min_interval' and 'max_interval'	invalid interval between linked activities
29	Can't link activities:9382904: add_appt_link: Circular appointment links are not allowed: typeID = 2, fromApptID = 3998006, toApptID = 3998007	link to be created is a circular (loop) link
30	Can't link activities:9382904: add_appt_link: Duplicate appointment links are not allowed: typeID = 2, fromApptID = 3998006, toApptID = 3998007	link to be created already exists
32	Too many concurrent requests	the maximum number of 'get_activities' requests has been exceeded
33	Request too expensive	the 'get_activities' request runs longer than the maximum time
100	Internal error	all other cases

5 Previous Versions

The Activity Management functionality in version 4.5 has not changed as compared to version 4.4 except the following:

5.1 User Authentication Node

The 'company' field now accepts the 'instance name' instead of the 'company' name.

5.2 New Methods

With the introduction of the Required Inventory entity, the Activity Management API now supports methods related to required inventory:

- 'set_required_inventories'
- 'get_required_inventories'

A new method allowing to search activities by multiple criteria has been implemented (available only in API Version 2):

- 'get_activities'

5.3 Activity Link-Related Methods Changed

The activity link-related methods have been changed to reflect changes to the activity link concept introduced in version 4.5. Refer to the corresponding section for more details.