TOA

TECHNOLOGIES

SmartLocation SDK

for

ETAdirect

Version 4.5

Part Number E62378-01

TOA Technologies, 2014

Table of Contents

1 Introduction

1.1 Document Purpose

This document is intended as the developer's guide for Location API of the ETAdirect Platform. It contains information which helps to understand the details of using API which the Platform provides. Please refer to this guide to ensure successful integration of third-party systems with ETAdirect SmartLocation services.

1.2Document Scope

The document provides a complete guide to the Location API with detailed descriptions of each method as well as examples and terminology which will help with writing a remote client for using this service online.

1.3Target Audience

This document is intended for developers who work on the integration of third-party systems with ETAdirect professional web services. This guide is a set of useful tips and examples illustrating how to use them easily.

1.4Glossary

Term	Explanation
API	Application Programming Interface – a particular set of rules and specifications that software programs follow to communicate and interact with each other
Company	1) Legal entity, using ETAdirect 2) Entity that represents a Client in ETAdirect system. The case-sensitive company identifier is created by TOA technologies during the process of implementation
ISO 8601 format	See http://en.wikipedia.org/wiki/ISO_8601

2 SmartLocation-Specific Terminology

In this document SmartLocation-specific terms and abbreviations are used as defined below.

Term	Explanation
login	case-sensitive user identifier that should be used to obtain the necessary privileges to access the protected resources of ETAdirect. The value provided by TOA Technologies during the integration process should be used as login.
password	case-sensitive set of characters protecting the user login. The password is provided by TOA Technologies during the integration process.
company	case-sensitive company identifier that represents a client in ETAdirect. The value provided by TOA Technologies during the integration process is to be used.
device	case-sensitive resource identifier such as external_id in terms of ETAdirect
longitude	geographic coordinate that specifies the east-west position of a point on the Earth surface. It is an angular measurement expressed in decimal degrees.
latitude	geographic coordinate that specifies the north-south position of a point on the Earth surface. It is an angular measurement expressed in decimal degrees.
time	retrieves a datetime object given as a string formatted according to a subset of the ISO-8601 standard. Please refer to the API documentation for full details or see the shortcuts* below.
src_entity	case-sensitive resource identifier. It may use such keys as uid, pid, external_id in terms of ETAdirect. Please note that the src_type field should refer to the proper type of field if different keys are used. These two properties are linked.
src_type	enumeration field that may contain the values only (user | resource | device). Please note that the corresponding types of this field require using the proper identifiers with the src_entity property. They should be linked as follows: uid<=>user | pid<=>resource | external_id<=>device. Other links are undefined, but for data to be displayed on the ETAdirect maps, the proper links must be used.
attributes	parameter allowing creation and storage of custom objects in addition to the predefined ones, such as the current location description. It also allows retrieving such custom data for each getter method.
zoom	integer field used as a filter option allowing to expand the track by points for which the distance in meters from the previous one is greater than this parameter. This parameter helps to scale the track for maps which have the ability of zooming in/out. The value set to (0) retrieves the full history of positioning with additional items such as points of idle. The max values give the best results of filtering the linear parts of the trace.
idletime	integer field the meaning of which may differ for requests and responses depending on the context. In the context of a request it is used as an acceptable threshold of

	idle time, but in the context of a response it means the actual time of idleness of the resource (seconds).
radius	integer field representing the radius of a single latitude / longitude point for a given location to search around. Please see the description of each method for the units of measurement.
limit	parameter allowing to set the limit of records used for filtering. When this parameter is omitted, the limiting is disabled. For example, this parameter can be used when only the first three resources nearest to the user's location are to be filtered. It is used in the same manner in all APIs where applicable.
coords	object that may be used for retrieving the resource location. It should contain such fields as longitude, latitude and time. Please refer to the terminology section for the details on each field.
history	array which may be used for retrieving the history of a given resource. See the get_source_history example for details. Each item of the array should contain such fields as longitude, latitude, time and attributes. Please refer to the terminology section for the details on each field.
track	array which may be used for retrieving the track for a given resource. See the get_source_track example for details. Each item of array should contain such required fields as longitude, latitude, time and attributes. One optional parameter such as idletime is supported. Please refer to the terminology section for the details on each field.
resources	array which may be used for retrieving the resources returned by the SmartLocation search engine. See the get_sources_in_area example for details. Each item of the array should contain such fields as resource, source, stype, longitude, latitude, time and distance. Please refer to the terminology section for the details on each field.
resource	case-sensitive source identifier of the resource such as pid in terms of ETAdirect. One of the unique features of ETAdirect SmartLocation is the ability to create unlimited many-to-many associations between entities and keep the history of changes when updated. The core of SmartLocation does not restrict the types of entities that may be associated, but the functional level of ETAdirect uses only such types as user | device | resource. Such entities as resources are the keys of all relationships in terms of ETAdirect which may be used in the search engine independently of context. They always return the source entity located and the key such as resource (pid) to which it relates. For example, ETAdirect SmartCollaboration / Who's Nearby service uses SmartLocation search engine in this manner.

source	case-sensitive source identifier for which the set_position method is used. Its meaning is the same context as for src_entity. Please note that not only web services may use ETAdirect SmartLocation and, therefore, it may return not only the identifier of device but also that of resource or user.
stype	enumeration field that may contain values only (user | resource | device). Please see src_type for details
distance	distance to the object (kilometers) determined when the ETAdirect SmartLocation search engine is requested to locate the nearby sources around a certain area
error_code	integer value that describes the status code which can be returned when a method is called
error_msg	string value returning the error message generated during execution
batch	ETAdirect SmartLocation technology that exposes a uniform service to operate on collection of API calls per single request and describes how to develop the support executing multiple operations sent in a single SOAP request through the use of Batching. Please read section Batch API to learn the details.

2.1 Time

The datetime object given as a string, formatted according to a subset of the ISO-8601 standard. If the representation of time indicates no UTC relation, the time is assumed to be the UTC time. External systems may use their time zone relation against UTC represented as ±hhmm. SmartLocation keeps time in UTC. This is important for requests of resource history with time points for each position.

Notes on the subset of ISO 8601 used in SmartLocation notation. Some formats of time described in ISO 8601 are irrelevant for SmartLocation, so the formal specification of time has been restricted.

```
YYYY-MM-DD
YYYY-MM-DD HH:MM:SS
YYYY-MM-DD HH:MM:SS
YYYY-MM-DD HH:MM:SS±hhmm
```

2.1.1 Combined date and time representations

The time designator [T] may be used to show the start of the time component of the representation. For example:

```
YYYY-MM-DD[T]HH:MM:SS±hhmm
```

2.1.2 Time zone designators

There are no time zone designators in ISO 8601. Time is only represented as local time or in relation to UTC.

If the representation of time indicates no UTC relation, the time is assumed to be the UTC time. This makes it fundamentally different from ISO 8601, because SmartLocation uses only time points of remote systems and no local time zone is defined specifically for these cases. The original quotation of ISO 8601 is as follows: 'While it may be safe to assume local time when communicating in the same time zone, it is ambiguous when used in communicating across different time zones. It is usually preferable to indicate a time zone (zone designator) using the standard notation.

2.1.3 UTC

If the time is shown in UTC, a (Z) should be added directly after the time without a space. (Z) is the zone designator for the zero UTC offset.

2.1.3.1 Time offsets from UTC

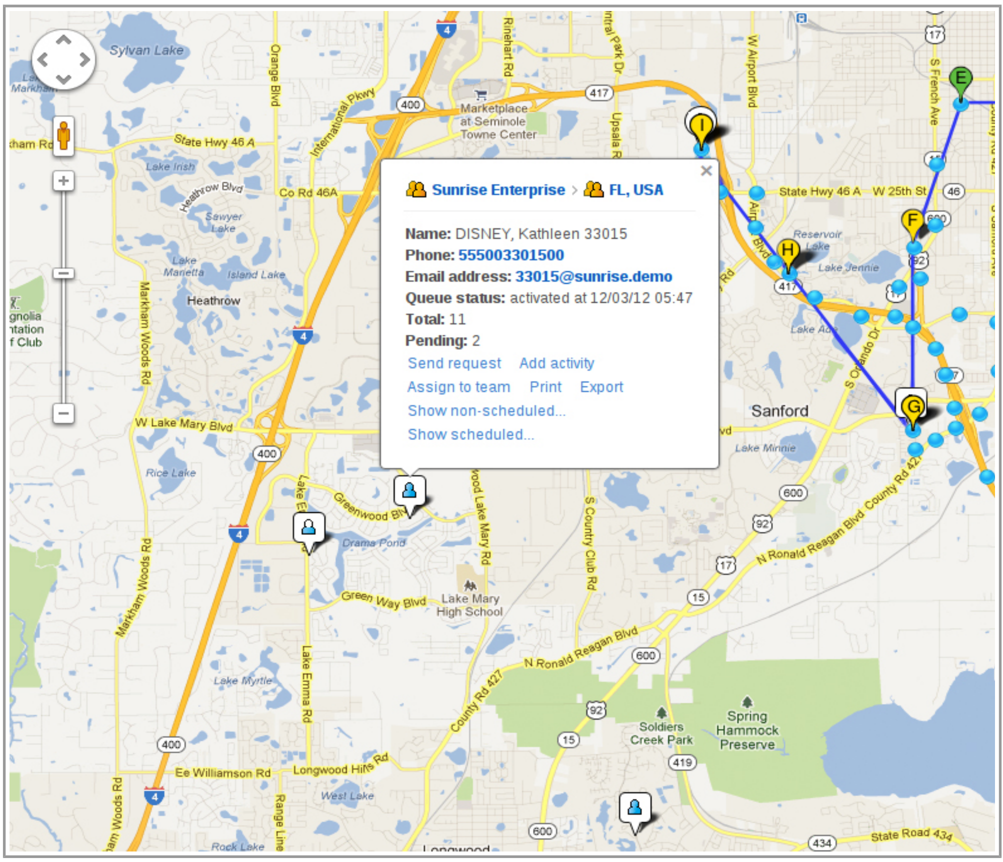
The offset from UTC is given in the format ±[hh][mm] only. This format, on one hand, may be used as the time zone offset from UTC and, on the other hand, is supported by most common libraries in the native mode.

The offset can also be used when the UTC time is known, while the local offset is not. In this case the offset is (-0000), which is semantically different from (Z) or (+0000), as these imply that UTC is the preferred reference point for those times.

3 API Overview

The Location API provides a simple and structured way to communicate with ETAdirect SmartLocation service. This API allows to remotely update ETAdirect Location-Based Services to change their storages with resource data from the third-party providers that is needed to search for locations. This API provides a complete set of methods to monitor the location / traces of resources the client uses for ETAdirect. Also, ETAdirect SmartLocation service exports the additional feature such as the get_sources_in_area method that allows retrieving the information about resources location within a certain area that can be described with the longitude, latitude, and radius parameters.

4 Authentication



Sunrise Enterprise > FL, USA

Name: DISNEY, Kathleen 33015
Phone: 555003301500
Email address: 33015@sunrise.demo
Queue status: activated at 12/03/12 05:47
Total: 11
Pending: 2

Figure 1: Map displaying resources location

Most of the resources in the ETAdirect APIs are protected by security privileges granted to a specific user. Each SOAP request to a protected Resource in ETAdirect API must be accompanied by one of the Authentication methods.

Location API Authentication passes the user’s credentials consisting of ETAdirect username and password to SOAP / user node for each request. All tags are mandatory. They are shown in the table below.

Name	Type	Required	Description
now	datetime	Yes	the current time Please refer to the terminology section for the details (the term 'time').
company	string	Yes	case-sensitive company identifier Please refer to the terminology section for the details.

Name	Type	Required	Description
login	string	Yes	case-sensitive user identifier Please refer to the terminology section for the details (the term 'time').
auth_string	string	Yes	Authentication hash. The formula is as follows: MD5(now + MD5(password)). Please refer to the terminology section for the details (the term 'password').

If authentication fails and access is denied, then the following SOAP Fault is returned.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <soapenv:Fault>
         <faultcode>soapenv:Client</faultcode>
         <faultstring>Authentication_Error</faultstring>
      </soapenv:Fault>
   </soapenv:Body>
</soapenv:Envelope>
```

5 API Reference

The API performs the following SOAP functions necessary for ETAdirect location services.

set_position	updates the service with the current location of resource. The object (attributes) is unspecified by its structure and may contain any parameters or objects within. This parameter is reserved to upload the custom location-related properties preferred by the client to the service.
get_position	retrieves the last location which was set for the resource with the set_position function. Please note that this function does not return the custom properties set with the attributes parameter because the additional API name get_position_attr is reserved.
get_position_attr	retrieves the custom properties which were set with the attributes parameter for the last location of the resource determined by the set_position function
get_source_history	returns the history of positioning for the given resource during the specific period of time
get_source_track	This function filters the history by applying the algorithm reducing the number of points in a curve that is approximated by a series of points. For details refer to http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm. The initial algorithm has been significantly updated and now supports the type of points (zoom) and (idle). (zoom) is the maximum distance between vertices of the curves that have been approximated (idle) is an additional attribute in seconds for points where the inactivity threshold was exceeded
get_sources_in_area	retrieves the information about the location of resources within the area defined by the (longitude) (latitude) and (radius) parameters

5.1 'set_position' Method

This method updates the service with the current location of resource. The longitude and latitude parameters are the coordinates that describe the actual location, while time is used to create the timestamp of the point in time when the position was taken. The additional attributes field provides the feature allowing to upload the custom location-related properties preferred by the client in free xml format.

5.1.1 'set_position' Request Parameters

Name	Type	Required	Description
company	string	Yes	case-sensitive company identifier Please refer to the terminology section for the details.
device	string	Yes	case-sensitive device identifier Please refer to the terminology section for the details.
longitude	double	Yes	geographic coordinate that specifies the east-west position of a point on the Earth surface. It is an angular measurement expressed in decimal degrees.
latitude	double	Yes	geographic coordinate that specifies the north-south position of a point on the Earth surface. It is an angular measurement expressed in decimal degrees.
time	datetime	Yes	timestamp of the point in time when the position was taken Please refer to the terminology section for the details.
attributes	object (xsd:any)	No	custom object Please refer to the terminology section for the details.

5.1.1.1 'set_position' Request Example

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:toa:location">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:set_position>
         <!--You may enter the following 7 items in any order-->
         <user>
            <!--You may enter the following 4 items in any order-->
            <now>2013-05-14 15:39:57+0000</now>
            <login>soap</login>
            <company>sunrise</company>
            <auth_string>ea34f387c8c12f3ae72a56ca902ac6b3</auth_string>
         </user>
         <company>sunrise</company>
         <device>100</device>
         <longitude>12.120000</longitude>
         <latitude>10.100000</latitude>
         <time>2013-05-14 15:37:57+0000</time>
         <!--Optional:-->
         <attributes>
            <!--You may enter ANY elements at this point-->
            <speed>79</speed>
            <altitude>100.1</altitude>
            <engine>
                <fuel>30.7</fuel>
                <rate>9.6</rate>
            </engine>
         </attributes>
      </urn:set_position>
   </soapenv:Body>
</soapenv:Envelope>
```

5.1.2 'set_position' Response Parameters

Name	Type	Required	Description
error_code	integer	Yes	Please refer to the terminology section for the details.

5.1.2.1 'set_position' Response Example

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:toa:location">
   <soapenv:Body>
      <urn:set_position_response>
         <error_code>0</error_code>
      </urn:set_position_response>
   </soapenv:Body>
</soapenv:Envelope>
```

5.2 'get_position' Method

This method retrieves the last location which was set for the resource with the set_position function. Please note that the function does not return the custom properties set with the attributes parameter because the additional API name get_position_attr is reserved.

5.2.1 'get_position' Request Parameters

Name	Type	Required	Description
company	string	Yes	case-sensitive company identifier Please refer to the terminology section for the details.
device	string	Yes	case-sensitive device identifier Please refer to the terminology section for the details.

5.2.1.1 'get_position' Request Example

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:toa:location">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:get_position>
         <!--You may enter the following 3 items in any order-->
         <user>
            <!--You may enter the following 4 items in any order-->
            <now>2013-05-14 15:39:57+0000</now>
            <login>soap</login>
            <company>sunrise</company>
            <auth_string>ea34f387c8c12f3ae72a56ca902ac6b3</auth_string>
         </user>
         <company>sunrise</company>
         <device>100</device>
      </urn:get_position>
   </soapenv:Body>
</soapenv:Envelope>
```

5.2.2 'get_position' Response Parameters

Name	Type	Required	Description
error_code	integer	Yes	Please refer to the terminology section for details.
coords	object	No	Please refer to the terminology section for details.

5.2.2.1 'get_position' Response Example

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:toa:location">
   <soapenv:Body>
      <urn:get_position_response>
         <error_code>0</error_code>
         <coords>
            <longitude>12.120000</longitude>
            <latitude>10.100000</latitude>
            <time>2013-05-14 15:37:57+0000</time>
         </coords>
      </urn:get_position_response>
   </soapenv:Body>
</soapenv:Envelope>
```

5.3 'get_position_attr' Method

This method retrieves the custom properties which were set with the attributes parameter for the last location of resource with the set_position function. This API name is reserved to effectively deliver the custom properties but with a different method covering the typical use-case of determining the resource location separately from reading the custom properties which is an extra scenario for it.

5.3.1 'get_position_attr' Request Parameters

Name	Type	Required	Description
company	string	Yes	case-sensitive company identifier Please refer to the terminology section for details.
device	string	Yes	case-sensitive device identifier Please refer to the terminology section for details.

5.3.1.1 'get_position_attr' Request Example

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:toa:location">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:get_position_attr>
         <!--You may enter the following 3 items in any order-->
         <user>
            <!--You may enter the following 4 items in any order-->
            <now>2013-05-14 15:39:57+0000</now>
            <login>soap</login>
            <company>sunrise</company>
            <auth_string>ea34f387c8c12f3ae72a56ca902ac6b3</auth_string>
         </user>
         <company>sunrise</company>
         <device>100</device>
      </urn:get_position_attr>
   </soapenv:Body>
</soapenv:Envelope>
```

5.3.2 'get_position_attr' Response Parameters

Name	Type	Required	Description
error_code	integer	Yes	Please refer to the terminology section for details.
attributes	object	No	Please refer to the terminology section for details.

5.3.2.1 'get_position_attr' Response Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:toa:location">
   <soapenv:Body>
      <urn:get_position_attr_response>
         <error_code>0</error_code>
         <attributes>
            <speed>79</speed>
            <altitude>100.1</altitude>
            <engine>
               <fuel>30.7</fuel>
               <rate>9.6</rate>
            </engine>
         </attributes>
      </urn:get_position_attr_response>
   </soapenv:Body>
</soapenv:Envelope>
```

5.4 'get_source_history' Method

This method returns the history of positioning for a given resource during the specified period of time. It does not use any filters, only the raw history of positioning for the resource returned.

5.4.1 'get_source_history' Request Parameters

Name	Type	Required	Description
company	string	Yes	case-sensitive company identifier Please refer to the terminology section for details
src_entity	string	Yes	case-sensitive resource identifier Please refer to the terminology section for details
src_type	enum	Yes	Please refer to the terminology section for details
time_from	datetime	Yes	filter option being the starting point of history to be retrieved Please refer to the terminology section for details (the term 'time').
time_to	datetime	Yes	filter option being the end point of history to be retrieved Please refer to the terminology section for details (the term 'time').

5.4.1.1 'get_source_history' Request Example

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:toa:location">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:get_source_history>
         <!--You may enter the following 6 items in any order-->
         <user>
            <!--You may enter the following 4 items in any order-->
            <now>2013-05-14 15:39:57+0000</now>
            <login>soap</login>
            <company>sunrise</company>
            <auth_string>ea34f387c8c12f3ae72a56ca902ac6b3</auth_string>
         </user>
         <company>sunrise</company>
         <src_entity>100</src_entity>
         <src_type>device</src_type>
         <time_from>2013-05-12 15:37:57+0000</time_from>
         <time_to>2013-05-14 15:40:57+0000</time_to>
      </urn:get_source_history>
   </soapenv:Body>
</soapenv:Envelope>
```

5.4.2 'get_source_history' Response Parameters

Name	Type	Required	Description
error_code	integer	Yes	Please refer to the terminology section for the details.
history	array	No	Please refer to the terminology section for the details.

5.4.2.1 'get_source_history' Response Example

```
  <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:toa:location">
     <soapenv:Body>
        <urn:get_source_history_response>
           <error_code>0</error_code>
           <history>
              <item>
                 <longitude>12.120000</longitude>
                 <latitude>10.100000</latitude>
                 <time>2013-05-14 15:08:17+0000</time>
                 <attributes/>
              </item>
              <item>
                 <longitude>12.120000</longitude>
                 <latitude>10.100000</latitude>
                 <time>2013-05-14 15:11:35+0000</time>
                 <attributes/>
              </item>
              <item>
                 <longitude>12.120000</longitude>
                 <latitude>10.100000</latitude>
                 <time>2013-05-14 15:19:54+0000</time>
                 <attributes/>
              </item>
              <item>
                 <longitude>12.120000</longitude>
                 <latitude>10.100000</latitude>
                 <time>2013-05-14 15:32:49+0000</time>
                 <attributes/>
              </item>
              <item>
                 <longitude>12.120000</longitude>
                 <latitude>10.100000</latitude>
                 <time>2013-05-14 15:37:57+0000</time>
                 <attributes>
                    <speed>79</speed>
                    <altitude>100.1</altitude>
                    <engine>
                       <fuel>30.7</fuel>
                       <rate>9.6</rate>
                    </engine>
                 </attributes>
              </item>
              <item>
                 <longitude>12.120000</longitude>
                 <latitude>10.100000</latitude>
                 <time>2013-05-14 15:39:57+0000</time>
                 <attributes/>
              </item>
           </history>
        </urn:get_source_history_response>
     </soapenv:Body>
</soapenv:Envelope>
```

5.5 'get_source_track' Method

This method retrieves the history of positioning for a given resource during the specified period of time, and applies a filter to reduce the number of points in a final approximated track which is to be returned.

5.5.1 'get_source_track' Request Parameters

Name	Type	Required	Description
company	string	Yes	case-sensitive company identifier Please refer to the terminology section for the details.
src_entity	string	Yes	case-sensitive resource identifier Please refer to the terminology section for the details.
src_type	enum	Yes	Please refer to the terminology section for the details.
time_from	datetime	Yes	filter option being the starting point of track to be retrieved Please refer to the terminology section for the details (the term 'time')
time_to	datetime	Yes	filter option being the end point of track to be retrieved Please refer to the terminology section for the details (the term 'time')
zoom	integer	Yes	filter option allowing to expand the track by points for which the distance in meters from the previous one is greater than the parameter value Please refer to the terminology section for the details.
idletime	integer	No	additional attribute determining the acceptable idle time threshold (seconds). Please refer to the terminology section for the details.

5.5.1.1 'get_source_track' Request Example

```
  <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:toa:location">
     <soapenv:Header/>
     <soapenv:Body>
        <urn:get_source_track>
           <!--You may enter the following 7 items in any order-->
           <user>
              <!--You may enter the following 4 items in any order-->
              <now>2013-05-14 15:39:57+0000</now>
              <login>soap</login>
              <company>sunrise</company>
              <auth_string>ea34f387c8c12f3ae72a56ca902ac6b3</auth_string>
           </user>
           <company>sunrise</company>
           <src_entity>100</src_entity>
           <src_type>device</src_type>
           <time_from>2013-05-12 15:37:57+0000</time_from>
           <time_to>2013-05-14 15:40:57+0000</time_to>
           <zoom>100</zoom> <!-- meters -->
           <idletime>1200</idletime> <!-- 20 minutes in seconds -->
        </urn:get_source_track>
     </soapenv:Body>
</soapenv:Envelope>
```

5.5.2 'get_source_track' Response Parameters

Name	Type	Required	Description
error_code	integer	Yes	Please refer to the terminology section for the details.
track	array	No	Please refer to the terminology section for the details.

5.5.2.1 get_source_track Response Example

```
  <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:toa:location">
     <soapenv:Body>
        <urn:get_source_track_response>
           <error_code>0</error_code>
           <track>
              <item>
                 <longitude>12.120000</longitude>
                 <latitude>10.100000</latitude>
                 <time>2013-05-14 15:08:17+0000</time>
                 <attributes/>
              </item>
              <item>
                 <longitude>12.120000</longitude>
                 <latitude>10.100000</latitude>
                 <time>2013-05-14 15:08:17+0000</time>
                 <idletime>1780</idletime>
                 <attributes/>
              </item>
              <item>
                 <longitude>12.120000</longitude>
                 <latitude>10.100000</latitude>
                 <time>2013-05-14 15:39:57+0000</time>
                 <attributes/>
              </item>
```

```
                 </track>
              </urn:get_source_track_response>
           </soapenv:Body>
      </soapenv:Envelope>
```

5.6 'get_sources_in_area' Method

The method retrieves information about the resources location within a certain area, specified by longitude, latitude, and radius parameters.

Note:

– The method considers the resources which have the telemetry enabled as well as the resources, which don't, but receive their coordinates through a teamwork.

– The method uses ETAdirect user visibility settings for filtering the results to be returned, so that the user sees only those resources which have been added in the 'Resources' list in the 'Add/Edit user' context.

5.6.1 'get_sources_in_area' Request Parameters

Name	Type	Required	Description
company	string	Yes	case-sensitive company identifier Please refer to the terminology section for details.
longitude	double	Yes	longitude coordinate of the center point of the area to be searched. Longitude is the geographic coordinate that specifies the east-west position of such point on the Earth surface. It is an angular measurement expressed in decimal degrees.
latitude	double	Yes	latitude coordinate of the center point of the area to be searched. Latitude is the geographic coordinate that specifies the north-south position of such point on the Earth surface. It is an angular measurement expressed in decimal degrees.
radius	integer	Yes	radius (in kilometers) of the circular area to be searched with the center being the point defined with the latitude and longitude coordinates. Please refer to the terminology section for details.
time	datetime	No	timestamp in the history when the nearby resources are to be searched. When no parameter value is set, the value of the last known location is used. Please refer to the terminology section for details.

limit	integer	No	parameter limiting the number of records used in filtering Please refer to the terminology section for details.

5.6.1.1 'get_sources_in_area' Request Example

```
  <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:toa:location">
     <soapenv:Header/>
     <soapenv:Body>
        <urn:get_sources_in_area>
           <!--You may enter the following 7 items in any order-->
           <user>
              <!--You may enter the following 4 items in any order-->
              <now>2013-05-14 17:39:57+0000</now>
              <login>soap</login>
              <company>sunrise</company>
              <auth_string>ea34f387c8c12f3ae72a56ca902ac6b3</auth_string>
           </user>
           <company>sunrise</company>
           <longitude>-81.305037</longitude>
           <latitude>28.796396</latitude>
           <radius>10.1</radius> <!-- in kilometers -->
           <!--Optional:-->
           <time>2013-05-14 15:39:57+0000</time>
           <limit>100</limit> <!-- max resources nearby to return -->
        </urn:get_sources_in_area>
     </soapenv:Body>
</soapenv:Envelope>
```

5.6.2 'get_sources_in_area' Response Parameters

Name	Type	Required	Description
error_code	integer	Yes	Please refer to the terminology section for details.
resources	array	No	Please refer to the terminology section for details.

5.6.2.1 get_sources_in_area Response Example

```
  <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:toa:location">
     <soapenv:Body>
        <urn:get_sources_in_area_response>
           <error_code>0</error_code>
           <resources>
              <item>
                 <resource>310005</resource>
                 <source>102</source>
                 <stype>device</stype>
                 <longitude>-81.340227</longitude>
                 <latitude>28.736205</latitude>
                 <time>2013-05-14 15:39:57+0000</time>
                 <distance>7.521000</distance>
              </item>
              <item>
                 <resource>310007</resource>
                 <source>103</source>
                 <stype>device</stype>
                 <longitude>-81.358080</longitude>
                 <latitude>28.743580</latitude>
                 <time>2013-05-14 15:39:57+0000</time>
                 <distance>7.824000</distance>
              </item>
              <item>
                 <resource>310003</resource>
                 <source>101</source>
                 <stype>device</stype>
                 <longitude>-81.313963</longitude>
                 <latitude>28.709259</latitude>
                 <time>2013-05-14 15:39:57+0000</time>
                 <distance>9.728000</distance>
              </item>
           </resources>
        </urn:get_sources_in_area_response>
     </soapenv:Body>
</soapenv:Envelope>
```

6 Batch API

The standard version of Location API is designed to simplify the grouping of several API calls together into a single SOAP request. It also includes a limited ability to retrieve data for multiple requests simultaneously.

Each network connection made by a client, results in a certain amount of overhead even for architectures with highest throughput. If the application needs an ability to access significant amount of data in one go, changes need to be made to several objects at once, it is often more efficient to batch queries rather than make multiple individual requests.

To enable this, Location API supports batching. Batching allows to pass instructions for several different methods within a single request. Operations that are related into a single request can be grouped because ETAdirect SmartLocation processes the operations sequentially by default, and it is obvious that Location API supports the attribute to force SmartLocation to process the operations at the same time. Once all operations have been completed, a consolidated response will be returned and the connection will be closed. One of the fundamentals for the service is that despite of the fact that the model of processing was used, the service guarantees that the order of answers is kept in accordance with the order of calls in the request.

The formal specification has been developed in order to facilitate the understanding of difference between structure for single vs. multiple calls per request. It uses the same manner of calls but a different name of the method (batch) and additional root element with the name of the method for each group of parameters. There are no limits on amount of calls for each method within a single batch request.

It is possible that one of requested operations may produce an error. One of the reasons for this to happen is invalid parameters to perform one of the requested operations. In this scenario, the batch API does not prevent the work for other operations in a batch and performs a similar response to the standard Location API, but encapsulated in the batch. Other requests within the batch should still be completed successfully and will be returned, as usual, with the error_code (0) and body of response if required.

6.1 Limits

There is only one limit that is specified on a server side of ETAdirect SmartLocation and it is the size limit of request a client can send to Location API to prevent the improper usage of effective concurrency on the server side. There are no more limits used for batching.

Note: #firewall_payload_size_max – specifies the maximum accepted payload size (bytes) of a client requested, as indicated by the header of protocol. If this parameter is different from default it should be used with caution, because it may significantly reduce the effective concurrency. In other words, this parameter is similar to the value of time slice, which is used in operation system to schedule the next task which may not be interrupted by another one.

The default limit is about 100 calls per single batch. It's strongly recommended to not change the default limit. The default limit may be changed ONLY for situations when the performance on the client side is critically low, however the secure idiom of TOA allows to configure up to 1000 calls per single batch as maximum.

6.2 Example

6.2.1 Batch request Example

```
  <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:toa:location">
     <soapenv:Header/>
     <soapenv:Body>
        <urn:batch>
           <user>
              <!--You may enter the following 4 items in any order-->
              <now>2013-05-14 17:39:57+0000</now>
              <login>soap</login>
              <company>sunrise</company>
              <auth_string>ea34f387c8c12f3ae72a56ca902ac6b3</auth_string>
           </user>
           <!-- Zero or more repetitions -->
           <set_position>
              <!--You may enter the following 6 items in any order-->
              <company>sunrise</company>
              <device>100</device>
              <longitude>-81.305037</longitude>
              <latitude>28.796396</latitude>
              <time>2013-05-14 15:39:57+0000</time>
              <!-- Optional: -->
              <attributes/>
           </set_position>
           <set_position>
              <!--You may enter the following 6 items in any order-->
              <company>sunrise</company>
              <device>101</device>
              <longitude>-83.305037</longitude>
              <latitude>22.796396</latitude>
              <time>2013-05-14 15:39:57+0000</time>
              <!-- Optional: -->
              <attributes/>
           </set_position>
           <get_sources_in_area>
              <company>sunrise</company>
              <longitude>-81.305037</longitude>
              <latitude>28.796396</latitude>
              <radius>10.1</radius> <!-- in kilometers -->
              <!--Optional:-->
              <time>2013-05-14 15:39:57+0000</time>
              <limit>100</limit> <!-- max resources nearby to return ➔
           </get_sources_in_area>
        </urn:batch>
     </soapenv:Body>
</soapenv:Envelope>
```

6.2.2 Batch Response Example

```
  <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:urn="urn:toa:location">
     <soapenv:Body>
        <urn:batch>
           <set_position_response>
              <error_code>0</error_code>
           </set_position_response>
           <set_position_response>
              <error_code>0</error_code>
           </set_position_response>
           <get_sources_in_area_response>
              <error_code>0</error_code>
              <resources>
                 <item>
                    <resource>310005</resource>
                    <source>102</source>
                    <stype>device</stype>
                    <longitude>-81.340227</longitude>
                    <latitude>28.736205</latitude>
                    <time>2013-05-14 15:39:57+0000</time>
                    <distance>7.521000</distance>
                 </item>
                 <item>
                    <resource>310007</resource>
                    <source>103</source>
                    <stype>device</stype>
                    <longitude>-81.358080</longitude>
                           <latitude>28.743580</latitude>
                    <time>2013-05-14 15:39:57+0000</time>
                    <distance>7.824000</distance>
                 </item>
                 <item>
                    <resource>310003</resource>
                    <source>101</source>
                    <stype>device</stype>
                    <longitude>-81.313963</longitude>
                    <latitude>28.709259</latitude>
                    <time>2013-05-14 15:39:57+0000</time>
                    <distance>9.728000</distance>
                 </item>
              </resources>
           </get_sources_in_area_response>
        </urn:batch>
     </soapenv:Body>
</soapenv:Envelope>
```

7 Error Handling

The examples above illustrate the status in the best case scenario. However, even if the code is absolutely correct, exceptions still occur from time to time, because other factors involved may fail. Below are the examples of situations when a perfectly fine code can lead to server errors:

- the client terminated the request early while the application was still reading the incoming data
- the database server was overloaded and could not handle the query
- a filesystem is full
- a hard drive crashed
- a backend server overloaded
- a programming error occurred in the library which is in use
- network connection between the server and another system failed

These are just some examples, other issues may occur.

As with all applications, the application should include at least basic error handling. It will grant a number of benefits to the application and its end-users.

The most significant benefit is the efficient recovery of the application from error states. An application should have at least the simplest form of error handling, otherwise its execution can be brought to a complete halt by an error returned by the service. Another benefit is that the application can display errors that are more meaningful to the end-user, who will probably be not aware of the errors that may occur.

When due to a problem a request (or a portion of a request) cannot be executed, the response includes the list of errors instead of normal business data. If the request is executed successfully, but a minor problem was found or an unexpected change was made, the response includes the normal business data along with a list of warnings.

Understanding the types of errors which can be returned by the service helps designing the application to handle such errors efficiently.

7.1 SOAP Faults

In SOAP, an error result is returned to the client as a SOAP Fault, with the HTTP response code 500. If no SOAP Faults have been received, this means that the request was successful. The ETAdirect SmartLocation SOAP fault code is comprised of the standard SOAP 1.1 fault code, that is, either Server or Client. The SOAP fault string element contains a generic, human readable error message without details which are written to the log.

When in doubt, use the support request only to obtain the details of the issue, but faults can occur within the integration process only. ETAdirect SmartLocation service has been designed to be strongly resistant to application level faults, which are possible only if the network infrastructure breaks.

Another important issue is to distinguish when ETAdirect SmartLocation should return a SOAP Fault and when a result object that has the error information. Most client languages may transparently convert SOAP faults into runtime exceptions. Therefore, the issue can be reduced to when to throw an exception instead of returning an error value. SOAP Faults should be used only in cases of **Bad_Request**, **Schema_Error** or **Service_Unavailable**.

Often SOAP API is not the native format of API for a service, but rather a compatible layer which is used to export REST API under the Web Service that can be described by WSDL (Web Services Definition Language), but provides REST API for core modules within security zone.

For SmartLocation SOAP API, some general types of SOAP faults are used for infrastructure errors.

FaultCode	FaultString	Description
Client	Authentication_Error	Please refer to the Authentication section for details.
Client	Bad_Request	This error may occur when the SOAP processor is not able to understand the request. Ensure that the data, as well as the request, is SOAP-compliant.
Client	Schema_Validation_Error	This error is returned when a Schema validation check fails. Typically, this may occur when the SOAP request does not correspond to the WSDL schema provided by ETAdirect Service Provider.

Server	Service_Unavailable	This error occurs only when the SmartLocation Core Service is not reachable. Please inform the TOA support team.
Server	Bad_Response	The response which was returned by the SmartLocation Core Service cannot be converted to a SOAP-compliant format. Please contact the TOA support team for details.
Server	Schema_Validation_Error	This error is returned when a Schema validation check fails. This occurs when a response from the SmartLocation Core Service does not correspond to the WSDL schema provided to the client. Please contact the TOA support team for details.
Server	Configuration_Error	This exception appears when a problem is found in the configuration of service. This failure generally means that some configuration parameters were set incorrectly. Please contact the TOA support team.
Server	Internal_Error	An unexpected condition was encountered while the server was attempting to perform the request. Please contact the TOA support team for details.

Example

If authentication fails and access is denied, the following SOAP Fault is returned:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <soapenv:Fault>
         <faultcode>soapenv:Client</faultcode>
         <faultstring>Authentication_Error</faultstring>
      </soapenv:Fault>
   </soapenv:Body>
</soapenv:Envelope>
```

7.2 Application Errors

Application errors occur due to problems with business data on the client side or on the server side.

Below are examples of situations that could cause application errors:

- the request contains an invalid combination of fields
- the request is missing a required field
- the request is valid, but a problem occurred in internal business logic while processing the data

When an application error occurs, ETAdirect SmartLocation returns an error_code (different from zero) in response. Application errors are returned as normal SOAP responses. Each error also provides a message (error_msg) that indicates the cause of the problem, unlike the normal response which only contains business data and no error_msg.

Example

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:toa:location">
   <soapenv:Body>
      <urn:set_position_response>
         <error_code>-1</error_code>
         <error_msg>unable to add properties</error_msg>
      </urn:set_position_response>
   </soapenv:Body>
</soapenv:Envelope>
```