Oracle Field Service Cloud
Integrating with Smart Location API
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>i</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Document Purpose</td>
<td>1</td>
</tr>
<tr>
<td>Scope of the Document</td>
<td>1</td>
</tr>
<tr>
<td>Target Audience</td>
<td>1</td>
</tr>
<tr>
<td>Glossary</td>
<td>1</td>
</tr>
<tr>
<td>2 Smart Location -Specific Terminology</td>
<td>3</td>
</tr>
<tr>
<td>Smart Location-Specific Terminology</td>
<td>3</td>
</tr>
<tr>
<td>3 Smart Location API Overview</td>
<td>7</td>
</tr>
<tr>
<td>Smart Location API Overview</td>
<td>7</td>
</tr>
<tr>
<td>4 Authentication</td>
<td>9</td>
</tr>
<tr>
<td>Authentication</td>
<td>9</td>
</tr>
<tr>
<td>5 Smart Location Methods Description</td>
<td>11</td>
</tr>
<tr>
<td>Smart Location Methods Description</td>
<td>11</td>
</tr>
<tr>
<td>'set_position' Method</td>
<td>12</td>
</tr>
<tr>
<td>'get_position' Method</td>
<td>13</td>
</tr>
<tr>
<td>'get_position_attr' Method</td>
<td>15</td>
</tr>
<tr>
<td>'get_source_history' Method</td>
<td>16</td>
</tr>
<tr>
<td>'get_source_track' Method</td>
<td>18</td>
</tr>
<tr>
<td>'get_sources_in_area' Method</td>
<td>20</td>
</tr>
<tr>
<td>6 Batch API</td>
<td>25</td>
</tr>
<tr>
<td>Batch</td>
<td>25</td>
</tr>
<tr>
<td>7 Error Handling</td>
<td>29</td>
</tr>
<tr>
<td>Error Handling</td>
<td>29</td>
</tr>
</tbody>
</table>
Preface

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

Using Oracle Applications

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

Documentation Accessibility

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

Contacting Oracle

Access to Oracle Support

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

Comments and Suggestions

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

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

Document Purpose

This document is a developer’s guide for the Smart Location API. The Smart Location API is available as part of the Oracle Field Service Smart Location Cloud Service subscription. It contains information to help you understand the details of using the API that the platform provides.

Note: If you have not subscribed to the Oracle Field Service Smart Location Cloud Service, please refer to the GPS API.

Scope of the Document

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.

Target Audience

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

Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface – a particular set of rules and specifications that software programs follow to communicate and interact with each other</td>
</tr>
</tbody>
</table>
| Company          | 1) Legal entity, using Oracle Field Service Cloud
|                  | 2) Entity that represents a Client in Oracle Field Service Cloud; company is created by Oracle during the process of implementation              |
| ISO 8601 format  | see http://en.wikipedia.org/wiki/ISO_8601                                                                                                                                 |
## Smart Location-Specific Terminology

The following table defines the Smart Location-specific terms and abbreviations.

<table>
<thead>
<tr>
<th>Term</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>login</td>
<td>Case-sensitive user identifier that should be used to obtain the necessary privileges to access the protected resources of OFSC. The value provided by Oracle during the integration process should be used as login.</td>
</tr>
<tr>
<td>password</td>
<td>Case-sensitive set of characters protecting the user login. The password is provided by Oracle during the integration process.</td>
</tr>
<tr>
<td>company</td>
<td>Case-sensitive company identifier that represents a client in OFSC. The value provided by Oracle during the integration process is to be used.</td>
</tr>
<tr>
<td>device</td>
<td>Case-sensitive resource identifier such as 'external_id' in terms of OFSC.</td>
</tr>
<tr>
<td>longitude</td>
<td>Geographic coordinate that specifies the east-west position of a point on the Earth surface. It is an angular measurement expressed in decimal degrees.</td>
</tr>
<tr>
<td>latitude</td>
<td>Geographic coordinate that specifies the north-south position of a point on the Earth surface. It is an angular measurement expressed in decimal degrees.</td>
</tr>
<tr>
<td>time</td>
<td>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 in the Time section.</td>
</tr>
<tr>
<td>src_entry</td>
<td>Case-sensitive resource identifier. It may use such keys as 'uid', 'pid', 'external_id' in terms of OFSC. 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.</td>
</tr>
<tr>
<td>src_type</td>
<td>Enumeration field that may contain the values only (user</td>
</tr>
<tr>
<td>attributes</td>
<td>Parameter allowing creation and storage of company-specific objects in addition to the predefined ones, such as the current location description. It also allows retrieving such company-specific data for each get method. Attributes such as accuracy, speed, and altitude are the most popular attributes.</td>
</tr>
<tr>
<td>zoom</td>
<td>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.</td>
</tr>
<tr>
<td>Term</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>idletime</td>
<td>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).</td>
</tr>
<tr>
<td>radius</td>
<td>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.</td>
</tr>
<tr>
<td>limit</td>
<td>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.</td>
</tr>
<tr>
<td>coords</td>
<td>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.</td>
</tr>
<tr>
<td>history</td>
<td>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.</td>
</tr>
<tr>
<td>track</td>
<td>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.</td>
</tr>
<tr>
<td>resources</td>
<td>Array which may be used for retrieving the resources returned by the Smart Location 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.</td>
</tr>
<tr>
<td>resource</td>
<td>Case-sensitive source identifier of the resource such as ‘pid’ in terms of OFSC. One of the unique features of OFSC Smart Location is the ability to create unlimited many-to many associations between entities and keep the history of changes when updated. The core of Smart Location does not restrict the types of entities that may be associated, but the functional level of OFSC uses only such types as user</td>
</tr>
<tr>
<td>source</td>
<td>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 OFSC Smart Location and, therefore, it may return not only the identifier of device but also that of resource or user.</td>
</tr>
<tr>
<td>stype</td>
<td>Enumeration field that may contain values only (user</td>
</tr>
<tr>
<td>distance</td>
<td>Distance to the object (kilometers) determined when the OFSC Smart Location search engine is requested to locate the nearby sources around a certain area</td>
</tr>
<tr>
<td>error_code</td>
<td>Integer value that describes the status code which can be returned when a method is called</td>
</tr>
<tr>
<td>error_msg</td>
<td>String value returning the error message generated during execution</td>
</tr>
</tbody>
</table>
### Smart Location - Specific Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch</td>
<td>OFSC Smart Location 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.</td>
</tr>
</tbody>
</table>

---

### 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. Smart Location 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 Smart Location notation: Some formats of time described in ISO 8601 are irrelevant for Smart Location, so the formal specification of time has been restricted.

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

#### 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
```

#### 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 Smart Location 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.

<table>
<thead>
<tr>
<th>UTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

#### 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 Smart Location API Overview

Smart Location API Overview

The Smart Location API provides a simple and structured way to communicate with OFSC Smart Location service. This API allows to remotely update OFSC Smart 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 OFSC. Also, OFSC Smart Location 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

The 'user' structure is used for the request authentication. The relevant error is returned if the authentication fails.

If you created a login policy to allow access for only certain IP addresses, the login policy is applicable to the APIs as well.

For example, you defined to allow requests only from IP address 110.0.133.185 for a User Type="API_User" and with login policy "API_login_policy". This implies that authentication fails for a user accessing the APIs from an IP address other than 110.0.133.18, though the login credentials are correct.

<table>
<thead>
<tr>
<th>Number</th>
<th>Login</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>now</td>
<td>is different from the current time on the server and this difference exceeds the predefined time-window (30 minutes by default)</td>
</tr>
<tr>
<td>2</td>
<td>company</td>
<td>cannot be found in the Oracle Field Service Cloud</td>
</tr>
<tr>
<td>3</td>
<td>login</td>
<td>cannot be found for this company</td>
</tr>
<tr>
<td>4</td>
<td>user</td>
<td>user with this 'login' is not authorized to use the current method</td>
</tr>
<tr>
<td>5</td>
<td>auth_string</td>
<td>is not equal to md5(now+md5(password)) or auth_string = SHA256(now + SHA256(password+SHA256(login))));</td>
</tr>
</tbody>
</table>

For example: 'now' = "2005-07-07T09:25:02+00:00" and password = "Pa$$w0rD" then md5 (password) = "06395148c998f3388e87f222bdf5c84b" concatenated string = "2005-0707T09:25:02+00:0006395148c998f3388e87f222bdf5c84b" auth_string should be: auth_string = "62469089f554d7a3b4ac9be3f29a989"

Otherwise authentication is successful and the request is processed further.
# 5 Smart Location Methods Description

## Smart Location Methods Description

The API performs the following SOAP functions necessary for OFSC Smart Location services.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>set_position</td>
<td>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 company-specific location related properties preferred by the client to the service.</td>
</tr>
<tr>
<td>get_position</td>
<td>retrieves the last location which was set for the resource with the ‘set_position’ function. Please note that this function does not return the company-specific properties set with the attributes parameter because the additional API name ‘get_position_attr’ is reserved.</td>
</tr>
<tr>
<td>get_position_attr</td>
<td>retrieves the company-specific properties which were set with the attributes parameter for the last location of the resource determined by the ‘set_position’ function</td>
</tr>
<tr>
<td>get_source_history</td>
<td>returns the history of positioning for the given resource during the specific period of time</td>
</tr>
</tbody>
</table>
| 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                                                                                           |
'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 company-specific location-related properties preferred by the client in free XML format.

'set_position' Request

The following table describes the 'set_position' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>company</td>
<td>Yes</td>
<td>string</td>
<td>case-sensitive company identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Please refer to the terminology section for the details.</td>
</tr>
<tr>
<td>device</td>
<td>Yes</td>
<td>string</td>
<td>case-sensitive device identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Please refer to the terminology section for the details.</td>
</tr>
<tr>
<td>longitude</td>
<td>Yes</td>
<td>double</td>
<td>geographic coordinate that specifies the east-west position of a point on</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>the Earth surface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>It is an angular measurement expressed in decimal degrees.</td>
</tr>
<tr>
<td>latitude</td>
<td>Yes</td>
<td>double</td>
<td>geographic coordinate that specifies the north-south position of a point</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>on the Earth surface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>It is an angular measurement expressed in decimal degrees.</td>
</tr>
<tr>
<td>accuracy</td>
<td>No</td>
<td>double</td>
<td>company-specific object</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sub-entity of attributes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Indicates how close the device’s calculated position is from the resource</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>location on the map. Expressed as a radius in meters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>When submitted, you can determine whether a resource is idle with more</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>precision.</td>
</tr>
<tr>
<td>time</td>
<td>Yes</td>
<td>datetime</td>
<td>timestamp of the point in time when the position was taken</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Please refer to the terminology section for the details.</td>
</tr>
<tr>
<td>attributes</td>
<td>No</td>
<td>object (xsd:any)</td>
<td>company-specific object</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Please refer to the terminology section for the details.</td>
</tr>
</tbody>
</table>
'set_position' Request Example

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:toa:location">
  <soapenv:Header/>
  <soapenv:Body>
    <set_position xmlns="urn:toa:location">
      <user>
        <now>2018-02-05T18:13:57+0200</now>
        <login>soap</login>
        <company>18a006</company>
        <auth_string>650dde65d175c778529fc9b48c508c12</auth_string>
      </user>
      <company>18a006</company>
      <device>33001</device>
      <longitude>-81.273273</longitude>
      <latitude>28.798798</latitude>
      <time>2018-02-05T18:13:57+0200</time>
      <!--Optional:-->
      <attributes>
        <accuracy>27</accuracy>
      </attributes>
      <set_position>
      </soapenv:Body>
  </soapenv:Envelope>
```

'set_position' Response

The following table describes the 'set_position' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_code</td>
<td>integer</td>
<td>Yes</td>
<td>Please refer to the terminology section for details.</td>
</tr>
</tbody>
</table>

'set_position' Response Example

```xml
  <SOAP-ENV:Body>
    <ns1:set_position_response>
      <error_code>0</error_code>
    </ns1:set_position_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'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 company-specific properties set with the attributes parameter because the additional API name 'get_position_attr' is reserved.
'get_position' Request

The following table describes the 'get_position' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>company</td>
<td>Yes</td>
<td>string</td>
<td>case-sensitive company identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Please refer to the terminology section for the details.</td>
</tr>
<tr>
<td>device</td>
<td>Yes</td>
<td>string</td>
<td>case-sensitive device identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Please refer to the terminology section for the details.</td>
</tr>
</tbody>
</table>

'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>
```

'get_position' Response

The following table describes the 'get_position' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_code</td>
<td>integer</td>
<td>Yes</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
<tr>
<td>coords</td>
<td>object</td>
<td>No</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
</tbody>
</table>

'get_position' Response Example

```
  <SOAP-ENV:Body>
    <urn:get_position_response>
      <error_code>0</error_code>
      <coords>
        <longitude>12.120000</longitude>
      </coords>
    </urn:get_position_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
'get_position_attr' Method

This method retrieves the company-specific 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 company-specific properties but with a different method covering the typical use-case of determining the resource location separately from reading the company-specific properties which is an extra scenario for it.

'get_position_attr' Request

The following table describes the 'get_position_attr' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>company</td>
<td>Yes</td>
<td>string</td>
<td>case-sensitive company identifier</td>
</tr>
<tr>
<td>device</td>
<td>Yes</td>
<td>string</td>
<td>case-sensitive device identifier</td>
</tr>
</tbody>
</table>

'get_position_attr' Request Example

```xml
<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-->
        <time>2013-05-14 15:39:57+0000</time>
        <company>sunrise</company>
        <device>100</device>
      </user>
    </urn:get_position_attr>
  </soapenv:Body>
</soapenv:Envelope>
```
'get_position_attr' Response

The following table describes the 'get_position_attr' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_code</td>
<td>integer</td>
<td>Yes</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
<tr>
<td>attributes</td>
<td>object</td>
<td>No</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
</tbody>
</table>

'get_position_attr' Response Example

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

'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.

'get_source_history' Request

The following table describes the 'get_source_history' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>company</td>
<td>Yes</td>
<td>string</td>
<td>case-sensitive company identifier</td>
</tr>
<tr>
<td>src_entity</td>
<td>Yes</td>
<td>string</td>
<td>case-sensitive resource identifier</td>
</tr>
</tbody>
</table>

Please refer to the terminology section for the details.
### Smart Location Methods Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>src_type</td>
<td>Yes</td>
<td>enum</td>
<td>Please refer to the terminology section for details.</td>
</tr>
<tr>
<td>time_from</td>
<td>Yes</td>
<td>datetime</td>
<td>filter option being the starting point of history to be retrieved Please refer to the terminology section for details (the term 'time').</td>
</tr>
<tr>
<td>time_to</td>
<td>Yes</td>
<td>datetime</td>
<td>filter option being the starting point of history to be retrieved Please refer to the terminology section for details (the term 'time').</td>
</tr>
</tbody>
</table>

**'get_source_history' Request Example**

```xml
<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>ea34f387ce8c12f3ae72a56ca902ac6b3</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>
```

**'get_source_history' Response**

The following table describes the 'get_source_history' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_code</td>
<td>integer</td>
<td>Yes</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
<tr>
<td>history</td>
<td>array</td>
<td>No</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
</tbody>
</table>

**'get_source_history' Response Example**

```xml
  <SOAP-ENV:Body>
    <nsl:get_source_history_response>
      <error_code>0</error_code>
      <history>
```

---

**Oracle Field Service Cloud**

Integrating with Smart Location API

Chapter 5

Smart Location Methods Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>src_type</td>
<td>Yes</td>
<td>enum</td>
<td>Please refer to the terminology section for details.</td>
</tr>
<tr>
<td>time_from</td>
<td>Yes</td>
<td>datetime</td>
<td>filter option being the starting point of history to be retrieved Please refer to the terminology section for details (the term 'time').</td>
</tr>
<tr>
<td>time_to</td>
<td>Yes</td>
<td>datetime</td>
<td>filter option being the starting point of history to be retrieved Please refer to the terminology section for details (the term 'time').</td>
</tr>
</tbody>
</table>

**'get_source_history' Request Example**

```xml
<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>ea34f387ce8c12f3ae72a56ca902ac6b3</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>
```

**'get_source_history' Response**

The following table describes the 'get_source_history' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_code</td>
<td>integer</td>
<td>Yes</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
<tr>
<td>history</td>
<td>array</td>
<td>No</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
</tbody>
</table>

**'get_source_history' Response Example**

```xml
  <SOAP-ENV:Body>
    <nsl:get_source_history_response>
      <error_code>0</error_code>
      <history>
```
'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.

'get_source_track' Request

The following table describes the 'get_source_track' request parameters.
### Smart Location Methods Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| company | Yes      | string   | case-sensitive company identifier  
Please refer to the terminology section for the details. |
| src_entity | Yes  | string   | case-sensitive resource identifier  
Please refer to the terminology section for the details. |
| src_type | Yes      | enum     | Please refer to the terminology section for the details.                  |
| time_from | Yes     | datetime | 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  | Yes      | datetime | filter option being the end point of track to be retrieved  
Please refer to the terminology section for the details (the term 'time') |
| zoom     | Yes      | integer  | 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 | No       | integer  | additional attribute determining the acceptable idle time threshold (seconds)  
Please refer to the terminology section for the details. |

### 'get_position' Request Example

```xml
<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>ea34f3b7c8c12f3ae72a56ca902ac6b3</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>
```
The following table describes the 'get_source_track' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_code</td>
<td>integer</td>
<td>Yes</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
<tr>
<td>track</td>
<td>array</td>
<td>No</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
</tbody>
</table>

'get_source_track' Response Example

```
  <SOAP-ENV:Body>
    <ns1: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>
    </ns1:get_source_track_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

'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: (1) 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. (2) The method uses OFSC 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.

'get_sources_in_area' Request

The following table describes the 'get_sources_in_area' request parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Required</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>company</td>
<td>Yes</td>
<td>string</td>
<td>case-sensitive company identifier</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Please refer to the terminology section for the details.</td>
</tr>
<tr>
<td>longitude</td>
<td>Yes</td>
<td>double</td>
<td>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 a point on the Earth’s surface. It is an angular measurement expressed in decimal degrees.</td>
</tr>
<tr>
<td>latitude</td>
<td>Yes</td>
<td>double</td>
<td>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 a point on the Earth’s surface. It is an angular measurement expressed in decimal degrees.</td>
</tr>
<tr>
<td>radius</td>
<td>integer</td>
<td>Yes</td>
<td>Radius (distance from center point to edge, in kilometers) of the circular area to be searched with the center being the point defined with the latitude and longitude coordinates. See the terminology section for details.</td>
</tr>
<tr>
<td>time</td>
<td>No</td>
<td>datetime</td>
<td>Time stamp in the history when nearby resources are to be searched. If not used: Returns last known resource locations within the specific radius, if a resource was in the search radius within the past 24 hours. Note that a resource may have a more recent location outside the radius. If used with value = &quot;X&quot;: Returns location of resources at within a minute of time X within the specified radius. Interval is [X – 1 minute to X]. Note that &quot;X&quot; means a specific point in time, not &quot;since X&quot;. See the terminology section for details.</td>
</tr>
<tr>
<td>limit</td>
<td>No</td>
<td>integer</td>
<td>Parameter limiting the number of records used in filtering. If not used: Returns locations of all resources that were within a specified radius. If used with value = &quot;Y&quot;: Returns locations of Y resources within a specified radius. Note: Resources returned in the response do not consist of Y nearest to center point; Y is a randomly selected count of resources to be returned. Once Y resources have been selected by the API, they are sorted by nearest distance to the center point. See the terminology section for details.</td>
</tr>
</tbody>
</table>

'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-->
<!--You may enter the following 4 items in any order-->
<user>
<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>

'get_sources_in_area' Response

The following table describes the 'get_sources_in_area' response parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>error_code</td>
<td>integer</td>
<td>Yes</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
<tr>
<td>resources</td>
<td>array</td>
<td>No</td>
<td>Please refer to the terminology section for the details.</td>
</tr>
</tbody>
</table>

'get_sources_in_area' Response Example

```xml
<SOAP-ENV: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>
</resources>
</urn:get_sources_in_area_response>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

22
<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>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
6 Batch API

Batch

The standard version of the Smart 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, the Smart 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 OFSC Smart Location processes the operations sequentially by default, and it is obvious that the Smart Location API supports the attribute to force Smart Location 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 Smart 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.

Limits

There is only one limit that is specified on a server side of OFSC Smart Location and it is the size limit of request a client can send to the Smart 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. 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 Oracle allows to configure up to 1000 calls per single batch as maximum.
Batch Request Example

```xml
<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>
```

Batch Response Example

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:toa:location">
  <soapenv:Header/>
  <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>
    </urn:batch>
  </soapenv:Body>
</soapenv:Envelope>
```
7 Error Handling

Error Handling

The examples in the previous section 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. Following 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.

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 OFSC Smart Location 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 as well as the 'detail' element describing the reason of the error. The error details 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. OFSC Smart Location 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 OFSC Smart Location 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 Smart Location SOAP API, some general types of SOAP faults are used for infrastructure errors.

<table>
<thead>
<tr>
<th>FaultCode</th>
<th>FaultString</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>Authentication_Error</td>
<td>Please refer to the Authentication section for details.</td>
</tr>
<tr>
<td>Client</td>
<td>Bad_Request</td>
<td>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.</td>
</tr>
<tr>
<td>Client</td>
<td>Schema_Validation_Error</td>
<td>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 OFSC Service Provider.</td>
</tr>
<tr>
<td>Server</td>
<td>Service_Unavailable</td>
<td>This error occurs only when the Smart Location Core Service is not reachable. Please inform the Oracle support team.</td>
</tr>
<tr>
<td>Server</td>
<td>Bad_Response</td>
<td>The response which was returned by the Smart Location Core Service cannot be converted to a SOAP-compliant format. Please contact the Oracle support team for details.</td>
</tr>
<tr>
<td>Server</td>
<td>Schema_Validation_Error</td>
<td>This error is returned when a Schema validation check fails. This occurs when a response from the Smart Location Core Service does not correspond to the WSDL schema provided to the client. Please contact the Oracle support team for details.</td>
</tr>
<tr>
<td>Server</td>
<td>Configuration_Error</td>
<td>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 Oracle support team.</td>
</tr>
<tr>
<td>Server</td>
<td>Internal_Error</td>
<td>An unexpected condition was encountered while the server was attempting to perform the request. Please contact the Oracle support team for details.</td>
</tr>
</tbody>
</table>

SOAP Fault Example

If a request field ('src_type' in this case) contains an invalid value, the following SOAP Fault is returned:

```xml
<?xml version="1.0"?>
     SOAP-ENV:Body
     SOAP-ENV:Fault
     faultcode>soapenv:Client</faultcode>
     faultstring>Bad_Request</faultstring>
     <detail>
     <reason>[get_source_history/src_type]: invalid (enum) value(resourc): valid range: [user,resource,device]</reason>
     </detail>
</SOAP-ENV:Fault>
</SOAP-ENV:Envelope>
```
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, OFSC Smart Location 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

```xml
<?xml version="1.0"?>
  xmlns:urn="urn:oracle">  
  <SOAP-ENV:Body>
    <urn:set_position_response>
      <error_code>-1</error_code>
      <error_msg>unable to add properties</error_msg>
    </urn:set_position_response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Error Handling and Recovery

In order to build a robust and reliable application, it is important to make sure that it responds to errors properly. When a problem occurs and a request cannot be executed, the response includes a type of error instead of normal business data. This section covers the information that will help recognize and determine the errors. The following table describes the 'get_position_attr' response parameters.

<table>
<thead>
<tr>
<th>Code</th>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>incorrect input params: `time[_from</td>
<td>_to](...)` format is not compliant</td>
</tr>
<tr>
<td>-1</td>
<td>incorrect input params: <code>time_from(...)</code> &gt; <code>time_to(...)</code></td>
<td>Invalid time range is set within the request throughout the history.</td>
</tr>
<tr>
<td>-1</td>
<td>incorrect input params: keys are empty</td>
<td>No key parameters have been set within the request (should not be empty).</td>
</tr>
<tr>
<td>-1</td>
<td>past or future data disabled</td>
<td>Past or future timestamps cannot be used for the updates. Service request can be sent to the support team to reset this restriction, if there is requirement to do so. Note: this can reflect on past data if the entered date is incorrect.</td>
</tr>
<tr>
<td>-1</td>
<td>incorrect params: <code>src_type(...)</code> unknown</td>
<td>Invalid enumeration value is used for this field. Please refer to terminology section for details on which values are allowed.</td>
</tr>
</tbody>
</table>

The [...] value indicates any value a user may enter within the field.
Oracle Product Abbreviations Keyword Map