

GEOTrac Web Services (GWS)_{v14}

Contents

Overview	7
Get Activity Message Web Service	7
Purpose	7
Input	7
Output:	8
Sample Input	8
Sample Response	8
RIndex (Reason Code Values)	9
Get Activity Message With Land Description Web Service	12
Purpose	12
Input	13
Output:	13
Sample Input	13
Sample Response	13
Get GEOForm Messages	15
Input	15
Output:	15
Sample Input	16
Sample Response	16
Get ECM Activity Message Web Service	17
Purpose	17
Input	18
Output:	18
Sample Input	18
Sample Response	19
Get Asset Tracker Activity Message Web Service	20
Purpose	20
Input	21
Output:	21
Sample Input	21
Sample Response	21

Get Land Description.....	22
Input:.....	22
Output.....	23
Sample Input.....	23
Sample Response.....	23
Geocode Land Description.....	23
Input:.....	23
Output.....	23
Sample Input.....	24
Sample Response.....	24
Get Oil Gas POI.....	24
Input:.....	24
Output.....	25
Sample Input.....	25
Sample Response.....	25
Get Asset Odometer.....	26
Input:.....	26
Output.....	26
Sample Input.....	26
Sample Response.....	26
Get Asset Hour Meter.....	27
Input:.....	27
Output.....	27
Sample Input.....	27
Sample Response.....	27
Reset Asset Odometer.....	28
Input:.....	28
Output.....	28
Sample Input.....	28
Sample Response.....	28
Reset Asset Hour Meter.....	29
Input:.....	29

Output.....	29
Sample Input.....	29
Sample Response.....	29
Get ECM Odometer Reading.....	30
Input:.....	30
Output.....	30
Sample Input.....	30
Sample Response.....	30
Get ECM Hour Meter Reading	32
Input:.....	32
Output.....	32
Sample Input.....	32
Sample Response.....	32
Get ECM Fuel Consumed Reading	33
Input:.....	33
Output.....	33
Sample Input.....	33
Sample Response.....	33
IFTA Web Service	34
Purpose	34
Input.....	34
Output:.....	34
Sample Input.....	34
Sample Response.....	36
IFTA By Time zone Web Service	36
Purpose	36
Input.....	36
Output:.....	37
Sample Input.....	37
Sample Response.....	38
IFTA By Day Web Service	38
Purpose	38

Input.....	38
Output:.....	39
Sample Input.....	39
Sample Response.....	40
IFTA By Day By Time zone Web Service.....	40
Purpose.....	40
Input.....	40
Output:.....	41
Sample Input.....	41
Sample Response.....	42
Engine Fault Code Web Service.....	43
Purpose.....	43
Input.....	43
Output:.....	43
Sample Input.....	43
Sample Response.....	45
Last Known Location.....	46
Input:.....	46
Output (for each vehicle):.....	46
Sample Input.....	46
Sample Response.....	47
GEOFence.....	47
Input:.....	47
Output (for each GEOFence):.....	47
Sample Input.....	48
Sample Response.....	49
Assets (GEOTrac GPS Equipment Assets).....	50
Input:.....	50
Output (for each GEOTrac asset):.....	50
Sample Input.....	52
Sample Response.....	52
Messaging.....	55

Send Text Message	55
Input:.....	55
Output.....	55
Sample Input.....	55
Sample Response	57
Get Text Message.....	58
Input.....	58
Output (for each message)	58
Sample Input.....	58
Sample Response	59
Drivers	61
Input.....	61
Output:.....	61
Sample Input.....	61
Sample Response	62
URL to Service	64

Overview

The purpose of this document is to describe in detail GEOTrac Web Services (GWS). GWS is a collection of Web Services that allow GEOTrac customers to consume specific location based data from their fleet of vehicles.

Included in this document are a brief description of the function of each of the services accompanied by a typical example web service call and its corresponding sample response.

All requests will require a Customer ID, User name, and Password in their SOAP header, and can optionally be secured over HTTPS.

Get Activity Message Web Service

Purpose

GEOTrac Activity Web Service will allow GEOTrac customers to consume their GPS message activity in near real-time. This service is also known as the Activity Forwarding Service.

The following are high level features of GEOTrac Activity Web Service:

- Given a subscriber Company Id, the client can retrieve a block of Activity Messages from the GEOTrac Activity Message database for that Company Id
- The client will have the ability to request all the data since the last polling interval by passing along the Last Message Id received. The Last Message Id will be a Big Int auto incrementing unique Id for each message received
- The number of Activity Messages returned is limited by the fixed size number of messages. The maximum number of activity messages is dependent on the number of modems and polling interval the customer has for their account. (Contact your GEOTrac technical contact to find out what your companies maximum record count will be)
- The Last Message Id will be returned in the response
- GEOTrac will keep a maximum of one calendar years' worth of GPS Activity Data in the GEOTrac Activity Message database for that Company Id. This is subject to change depending on the type of service you have with GEOTrac.
- Client will be able to make multiple requests for the same data records.
- Authentication will be controlled by a username and password defined in the SOAP Header. Optional protected by using HTTPS.
- The recommended polling interval will be every 5 minutes. However this is dependent on the device messaging interval (how frequently does it send messages) and the number of devices within the customer's fleet.

Input

- Last Retrieved ID

Output:

- Message ID
- GEOTrac Message ID
- Message Time (UTC time)
- Modem Serial Number
- Customer ID
- Asset Serial Number
- Asset Unit Number
- *RIndex (number referenced reason code)
- Latitude (deg)
- Longitude (deg)
- Heading (deg)
- Speed (kmh)
- Alert Status (true = Is in Alert)
- Odometer Value (km)

Sample Input

```
<soap:Envelope>
  <soap:Header>
    <Authentication>
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
      <Password>string</Password>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <GetActivityMessage>
      <lastRetrievedMsgID>long</lastRetrievedMsgID>
    </GetActivityMessage>
  </soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetActivityMessageResponse >
      <GetActivityMessageResult>
        <LastMsgID>long</LastMsgID>
        <Messages>
          <Message>
            <MsgId>long</MsgId>
            <GEOTracMsgID>guid</GEOTracMsgID>
```



```

<MsgTime>dateTime</MsgTime>
<ModemSN>string</ModemSN>
<CustomerId>int</CustomerId>
<AssetSN>string</AssetSN>
<AssetUN>string</AssetUN>
<RIndex>int</RIndex>
<Latitude>double</Latitude>
<Longitude>double</Longitude>
<Heading>double</Heading>
<Speed>int</Speed>
<IsAlert>boolean</IsAlert>
<OdometerValue>double</OdometerValue>
</Message>
<Message>
  <MsgId>long</MsgId>
  <MsgTime>dateTime</MsgTime>
  <ModemSN>string</ModemSN>
  <CustomerId>int</CustomerId>
  <AssetSN>string</AssetSN>
  <AssetUN>string</AssetUN>
  <RIndex>int</RIndex>
  <Latitude>double</Latitude>
  <Longitude>double</Longitude>
  <Heading>double</Heading>
  <Speed>int</Speed>
  <IsAlert>boolean</IsAlert>
  <OdometerValue>double</OdometerValue>
</Message>
</Messages>
</GetActivityMessageResult>
</GetActivityMessageResponse>
</soap:Body>
</soap:Envelope>

```

RIndex (Reason Code Values)

<i>RIndex</i>	<i>Reason</i>
1	Modem Test
2	Speed Limit Exceeded
6	No GPS Signal
8	Start
9	Schedule message
13	Stop

14	Ignition On
15	Ignition Off
16	Driver log message
17	MDT text message
18	Distress message
20	GPS Unplugged
21	Position Report
22	On-Road/Off-Road Indicator
23	Heart Beat
25	Perf. Button On
26	Perf. Button Off
42	PSM ACK
43	PSM NACK
44	PSM Man Down
45	PSM Out Of Range
46	PSM No Motion
47	PSM On
48	PSM Off
49	PSM DEV ADD
50	PSM DEV RMV
51	PSM DEV LST
52	Excessive Speeding
53	Stop Debounce
54	GPS Debounce
55	IGN OFF Debounce
56	Reserved
57	Reserved
59	PSM Back In Range
60	Aggressive Acceleration
61	Harsh Breaking
62	PSM Alert Cancellation
63	Seatbelt Off
66	OOTN Session Started
67	MDT Login
68	Health message
69	PSM Motion Sensor On
70	PSM Motion Sensor Off
80	Driver Signed In
81	No Driver Assigned
82	Driver Signed Out

85	Axle Down
86	Axle Up
87	PTO On
88	PTO Off
89	Tank Over Fill
90	Tank Level Adjusted
91	GPS Position Report
96	PSM Fall Panic
100	ASPS Load On
101	ASPS Load Off
103	Idle Heartbeat
11708057	1708 Fault Code Message
11939059	CAN J1939 Fault Code Message
11979058	CAN Fault Code Message
14000190	RPM Breach (J1708)
14200004	Engine Load Breach
14200005	Engine Coolant Temp
14200011	Intake Manifold Pressure
14200012	RPM Breach
14200013	Excessive Speeding
14200015	Intake Air Temp
14200017	Throttle position Breach
14200070	Ambient Temp
14200092	Engine Oil Temp
14200094	Engine Fuel Rate
14310001	Aggressive Acceleration
14310002	Harsh Braking
14310003	Collision Detected
14310004	Rough Driving
14310005	Seatbelt Off
14310010	MIL Lamp On
14310011	Red Stop Lamp On
14310012	Amber Warning Lamp On
15100000	MIM Restart
16000190	RPM Breach (J1939)
19900000	Unauthorized Driver Signed In
19900001	Expired Driver Signed In
26000235	Total Idle Hours
26000236	Total Idle Fuel Used (L)
26000245	Total Vehicle Distance (km)

26000246	Total Vehicle Hours
26000247	Total Engine Hours
26000250	Total Fuel Used (L)
50000000	DM Eligibility
50000001	DM Monthly USAGE
50000002	DM Monthly Data Limit Reached
50000003	IRD Link Check

Get Activity Message With Land Description Web Service

Purpose

GEOTrac Activity With Land Description Web Service will allow GEOTrac customers to consume their GPS message activity in near real-time. This service is also known as the Activity Forwarding Service.

The following are high level features of GEOTrac Activity With Land Description Web Service:

- Given a subscriber Company Id, the client can retrieve a block of Activity Messages from the GEOTrac Activity Message database for that Company Id
- The client will have the ability to request all the data since the last polling interval by passing along the Last Message Id received. The Last Message Id will be a Big Int auto incrementing unique Id for each message received
- The number of Activity Messages returned is limited by the fixed size number of messages. The maximum number of activity messages is dependent on the number of modems and polling interval the customer has for their account. (Contact your GEOTrac technical contact to find out what your companies maximum record count will be)
- The Last Message Id will be returned in the response
- GEOTrac will keep a maximum of one calendar years' worth of GPS Activity Data in the GEOTrac Activity Message database for that Company Id. This is subject to change depending on the type of service you have with GEOTrac.
- Client will be able to make multiple requests for the same data records.
- Authentication will be controlled by a username and password defined in the SOAP Header. Optional protected by using HTTPS.
- The recommended polling interval will be every 5 minutes. However this is dependent on the device messaging interval (how frequently does it send messages) and the number of devices within the customer's fleet.

Input

- Last Retrieved ID

Output:

- Message ID
- GEOTrac Message ID
- Message Time (UTC time)
- Modem Serial Number
- Customer ID
- Asset Serial Number
- Asset Unit Number
- RIndex (number referenced reason code)
- Latitude (deg)
- Longitude (deg)
- Heading (deg)
- Speed (kmh)
- Alert Status (true = Is in Alert)
- Odometer Value (km)
- Land Description

Sample Input

```
<soap:Envelope>
  <soap:Header>
    <Authentication>
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
      <Password>string</Password>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <GetActivityMessage>
      <lastRetrievedMsgID>long</lastRetrievedMsgID>
    </GetActivityMessage>
  </soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetActivityWithLandDescriptionResponse xmlns="http://assetcontrolcenter.com/">
      <GetActivityWithLandDescriptionResult>
        <LastMsgID>long</LastMsgID>
      </GetActivityWithLandDescriptionResult>
    </GetActivityWithLandDescriptionResponse>
  </soap:Body>
</soap:Envelope>
```

```

<Messages>
  <MessageLd>
    <MsgId>long</MsgId>
    <GEOTracMsgID>guid</GEOTracMsgID>
    <MsgTime>dateTime</MsgTime>
    <ModemSN>string</ModemSN>
    <CustomerId>int</CustomerId>
    <AssetSN>string</AssetSN>
    <AssetUN>string</AssetUN>
    <RIndex>int</RIndex>
    <Latitude>double</Latitude>
    <Longitude>double</Longitude>
    <Heading>double</Heading>
    <Speed>int</Speed>
    <IsAlert>boolean</IsAlert>
    <OdometerValue>double</OdometerValue>
    <LandDescription>string</LandDescription>
  </MessageLd>
  <MessageLd>
    <MsgId>long</MsgId>
    <GEOTracMsgID>guid</GEOTracMsgID>
    <MsgTime>dateTime</MsgTime>
    <ModemSN>string</ModemSN>
    <CustomerId>int</CustomerId>
    <AssetSN>string</AssetSN>
    <AssetUN>string</AssetUN>
    <RIndex>int</RIndex>
    <Latitude>double</Latitude>
    <Longitude>double</Longitude>
    <Heading>double</Heading>
    <Speed>int</Speed>
    <IsAlert>boolean</IsAlert>
    <OdometerValue>double</OdometerValue>
    <LandDescription>string</LandDescription>
  </MessageLd>
</Messages>
</GetActivityWithLandDescriptionResult>
</GetActivityWithLandDescriptionResponse>
</soap:Body>
</soap:Envelope>

```

Get GEOForm Messages

GEOForms requires the use of GEOTrac's "GEOForms Plugin" on the GEOTrac TABLET display. All of the runtime form data will be submitted to GEOTrac's Servers and made available to be downloaded in near real time using this Web Method Get GEOForms Messages.

The GetGEOFormMessages web service retrieves a list of D2Link messages (Position, Form, and Text) that have been converted and retained as GEOForm records.

The following are high level features of the GEOForms Web Service:

- Authentication will be controlled by a username and password defined in the SOAP Header. Optional protected by using HTTPS.
- Given a subscriber Company Id, the client can retrieve a list of that company's messages received by GEOTrac.
- The client specifies the Id of the last retrieved message. The service returns a list of messages with higher message Ids (a Long/BigInt value). Effectively, the client is requesting all newer messages than the message with the supplied id. Specifying an id of 0 (zero) retrieves *all* existing messages.
- The number of GEOForm Messages returned in one request is limited by the fixed size number of messages. (controlled by GEOTrac)
- The Id of the last returned message is identified in the response.
- The Client may make multiple requests for the same data records by specifying the same Id on future requests.
- The recommended polling interval will be every 5 minutes. However this is dependent on the device messaging interval (how frequently does it send messages) and the number of devices within the customer's fleet.
-

Input

- Previously retrieved Id (i.e. return messages with higher Ids than this)

Output:

- Id of last (latest) message returned in this service call
- For each returned message (limited to maximum 5,000 per invocation):
- Customer Id
 - Id of sending device
 - Asset Unit Number of asset containing the device
 - Unique message Id
 - Date/time (GMT) of message creation
 - Date/time (GMT) of message receipt at GEOTrac
 - Type of message (geospatial Position, Text, Form)
 - Message Subject

- Message Status (New, Open, In Progress, On Hold, Maintenance Due, Closed) as specified on the Asset Control Center's GEOForms website (all messages are originally New)
- Any remarks appended to the message in Asset Control Center's GEOForms website, including the remark Text, date/time of creation, and author
- Any fields associated with the message, include a field sequence number, the field name, and value

Sample Input

```
<soap:Envelope xmlns="http://assetcontrolcenter.com/" />
  <soap:Header>
    <Authentication xmlns="http://assetcontrolcenter.com/">
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
      <Password>string</Password>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <GetGEOFormMessages>
      <lastRetrievedMsgID>long</lastRetrievedMsgID>
    </GetGEOFormMessages>
  </soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetGEOFormMessagesResponse xmlns="http://assetcontrolcenter.com/">
      <GetGEOFormMessagesResult>
        <LastMsgId>long</LastMsgId>
        <DataList>
          <GEOFormData>
            <CustomerId>int</CustomerId>
            <DeviceId>guid</DeviceId>
            <AssetUnitNumber>string</AssetUnitNumber>
            <MsgId>long</MsgId>
            <DateCreated>datetime</DateCreated>
            <DateReceived>datetime</DateReceived>
            <Type>string</Type>
            <Subject>string</Subject>
            <Status>string</Status>
            <Remarks>
              <GEOFormAddendum>
```



```

        <Addendum>string</Addendum>

<AddendumCreatedOn>datetime</AddendumCreatedOn>

<AddendumCreatedBy>string</AddendumCreatedBy>
    </GEOFormAddendum>
    ...
    </Remarks>
    <Values>
        <GEOFormField>
            <Sequence>int</Sequence>
            <Name>string</Name>
            <Value>string</Value>
        </GEOFormField>
        ...
    </Values>
</GEOFormData>
... (Repeats)
</DataList>
</GetGEOFormMessagesResult>
</GetGEOFormMessagesResponse>
</soap:Body>
</soap:Envelope>

```

Get ECM Activity Message Web Service

Purpose

GEOTrac ECM Data Web Service will allow GEOTrac customers to consume their ECM message in near real-time. This service is also known as the ECM Data Forwarding Service.

The following are high level features of GEOTrac ECM Data Web Service:

- Given a subscriber Company Id, the client can retrieve a block of ECM Data from the GEOTrac ECM messages database for that Company Id
- The client will have the ability to request all the data since the last polling interval by passing along the Last Message Id received. The Last Message Id will be a Big Int auto incrementing unique Id for each message received

- The number of ECM Messages returned is limited by the fixed size number of messages. The maximum number of ECM messages is dependent on the number of modems and polling interval the customer has for their account. (Contact your GEOTrac technical contact to find out what your companies maximum record count will be)
- The Last Message Id will be returned in the response
- GEOTrac will keep a maximum of one calendar years' worth of ECM Data in the GEOTrac ECM Message database for that Company Id. This is subject to change depending on the type of service you have with GEOTrac.
- Client will be able to make multiple requests for the same data records.
- Authentication will be controlled by a username and password defined in the SOAP Header. Optional protected by using HTTPS.
- The recommended polling interval will be every 5 minutes. However this is dependent on the device messaging interval (how frequently does it send messages) and the number of devices within the customer's fleet.

Input

- Last Retrieved ID

Output:

- Message ID
- GEOTrac Message ID (MsgGuid)
- Message Time (UTC time)
- Modem Serial Number
- Customer ID
- Asset Serial Number
- Asset Unit Number
- RIndex (number referenced reason code)
- DTC Protocol
- DTC Code
- DTC Value
- MIM Value (Post calculated value)
- MIM Threshold Value (Post calculated value)
- MIM Info Data
- ECM Msg Category

Sample Input

```
<soap:Envelope>
  <soap:Header>
    <Authentication>
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
```

```

    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
  <GetEcmActivityMessage xmlns="http://assetcontrolcenter.com/">
    <lastRetrievedMsgID>long</lastRetrievedMsgID>
  </GetEcmActivityMessage>
</soap:Body>
</soap:Envelope>

```

Sample Response

```

<soap:Envelope>
  <soap:Body>
    <GetEcmActivityMessageResponse xmlns="http://assetcontrolcenter.com/">
      <GetEcmActivityMessageResult>
        <LastMsgID>long</LastMsgID>
        <EcmDataList>
          <EcmData>
            <MsgId>long</MsgId>
            <MsgTime>dateTime</MsgTime>
            <ModemSN>string</ModemSN>
            <CustomerId>int</CustomerId>
            <AssetUN>string</AssetUN>
            <MsgGuid>guid</MsgGuid>
            <RIndex>int</RIndex>
            <DtcProtocol>int</DtcProtocol>
            <DtcCode>string</DtcCode>
            <DtcValue>int</DtcValue>
            <MimValue>double</MimValue>
            <MimThresholdValue>double</MimThresholdValue>
            <MimInfoData xsi:nil="true" />
            <EcmMsgCategory>Dtc or Mim</EcmMsgCategory>
          </EcmData>
          <EcmData>
            <MsgId>long</MsgId>
            <MsgTime>dateTime</MsgTime>
            <ModemSN>string</ModemSN>
            <CustomerId>int</CustomerId>
            <AssetUN>string</AssetUN>
            <MsgGuid>guid</MsgGuid>
            <RIndex>int</RIndex>
            <DtcProtocol>int</DtcProtocol>
            <DtcCode>string</DtcCode>
            <DtcValue>int</DtcValue>
            <MimValue>double</MimValue>
            <MimThresholdValue>double</MimThresholdValue>
          </EcmData>
        </EcmDataList>
      </GetEcmActivityMessageResult>
    </GetEcmActivityMessageResponse>
  </soap:Body>
</soap:Envelope>

```

```
<MimInfoData xsi:nil="true" />
  <EcmMsgCategory>Dtc or Mim</EcmMsgCategory>
</EcmData>
</EcmDataList>
</GetEcmActivityMesssageResult>
</GetEcmActivityMesssageResponse>
</soap:Body>
</soap:Envelope>
```

Get Asset Tracker Activity Message Web Service

Purpose

GEOTrac Asset Tracker Data Web Service will allow GEOTrac customers to consume their Asset Tracker battery, engine run time and other engineering message in near real-time.

The following are high level features of GEOTrac Asset Tracker Data Web Service:

- Given a subscriber Company Id, the client can retrieve a block of Asset Tracker Data from the GEOTrac Asset Tracker messages database for that Company Id
- The client will have the ability to request all the data since the last polling interval by passing along the Last Message Id received. The Last Message Id will be a Big Int auto incrementing unique Id for each message received
- The number of Asset Tracker Messages returned is limited by the fixed size number of messages. The maximum number of Asset Tracker messages is dependent on the number of modems and polling interval the customer has for their account. (Contact your GEOTrac technical contact to find out what your companies maximum record count will be)
- The Last Message Id will be returned in the response.
- GEOTrac will keep a maximum of one calendar years' worth of Asset Tracker Data in the GEOTrac Asset Tracker Message database for that Company Id. This is subject to change depending on the type of service you have with GEOTrac.
- Client will be able to make multiple requests for the same data records.

- Authentication will be controlled by a username and password defined in the SOAP Header. Optional protected by using HTTPS.
- The recommended polling interval will be every 5 minutes. However this is dependent on the device messaging interval (how frequently does it send messages) and the number of devices within the customer's fleet.

Input

- Last Retrieved ID

Output:

- Message ID - BigInt
- GEOTrac Message ID - Message Guid ID
- Message Time (UTC time)
- Modem Serial Number
- Asset Unit Number
- RIndex - number referenced reason code, 68 is Health message
- Battery
- Engine Runtime display value
- GPS Failures - Count of failed GPS attempts since last Engineering Message
- GPS Mean Search Time - Average time for GPS search/acquisition
- Termination Reset - Toggles each time the termination count is reset
- Transmissions - Number of Satellite Transmissions since the device was enabled

Sample Input

```
<soap:Envelope>
  <soap:Header>
    <Authentication>
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
      <Password>string</Password>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <GetAssetTrackerActivityMessage xmlns="http://assetcontrolcenter.com/">
      <lastRetrievedMsgID>long</lastRetrievedMsgID>
    </GetAssetTrackerActivityMessage> </soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetAssetTrackerActivityMessageResponse xmlns="http://assetcontrolcenter.com/">
```

```

<GetAssetTrackerActivityMessageResult>
  <LastMsgID>long</LastMsgID>
  <AssetTrackerDataList>
    <AssetTrackerData>
      <MsgId>long</MsgId>
      <GEOTracMsgID>guid</GEOTracMsgID>
      <MsgTime>dateTime</MsgTime>
      <ModemSN>string</ModemSN>
      <AssetUN>string</AssetUN>
      <RIndex>int</RIndex>
      <Battery>double</Battery>
      <EngineRuntime>double</EngineRuntime>
      <GpsFailures>double</GpsFailures>
      <GpsMeanSearchTime>double</GpsMeanSearchTime>
      <TerminationReset>boolean</TerminationReset>
      <Transmissions>double</Transmissions>
    </AssetTrackerData>
    <AssetTrackerData>
      <MsgId>long</MsgId>
      <GEOTracMsgID>guid</GEOTracMsgID>
      <MsgTime>dateTime</MsgTime>
      <ModemSN>string</ModemSN>
      <AssetUN>string</AssetUN>
      <RIndex>int</RIndex>
      <Battery>double</Battery>
      <EngineRuntime>double</EngineRuntime>
      <GpsFailures>double</GpsFailures>
      <GpsMeanSearchTime>double</GpsMeanSearchTime>
      <TerminationReset>boolean</TerminationReset>
      <Transmissions>double</Transmissions>
    </AssetTrackerData>
  </AssetTrackerDataList>
</GetAssetTrackerActivityMessageResult>
</GetAssetTrackerActivityMessageResponse>
</soap:Body>
</soap:Envelope>

```

Get Land Description

Input:

- Longitude
- Latitude

Output:

- Land Description

Sample Input

```
<soap:Envelope>
<soap:Header>
  <Authentication>
    <CustomerID>string</CustomerID>
    <UserName>string</UserName>
    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
<GetLandDescription xmlns="http://assetcontrolcenter.com/">
  <longitude>double</longitude>
  <latitude>double</latitude>
</GetLandDescription>
</soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
<soap:Body>
<GetLandDescriptionResponse xmlns="http://assetcontrolcenter.com/">
  <GetLandDescriptionResult>string</GetLandDescriptionResult>
</GetLandDescriptionResponse>
</soap:Body>
</soap:Envelope>
```

Geocode Land Description

Input:

- Land Description

Output:

- Longitude
- Latitude

- Province

Sample Input

```
<soap:Envelope>
<soap:Header>
  <Authentication>
    <CustomerID>string</CustomerID>
    <UserName>string</UserName>
    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
<GeocodeLandDescription xmlns="http://assetcontrolcenter.com/">
  <landDescription>string</landDescription>
</GeocodeLandDescription> </soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
<soap:Body>
  <GeocodeLandDescriptionResponse xmlns="http://assetcontrolcenter.com/">
    <GeocodeLandDescriptionResult>
      <Longitude>double</Longitude>
      <Latitude>double</Latitude>
      <ProvinceState>string</ProvinceState>
    </GeocodeLandDescriptionResult>
  </GeocodeLandDescriptionResponse>
</soap:Body>
</soap:Envelope>
```

Get Oil Gas POI

Input:

- Latitude
- Longitude,
- Distance From Center
- Name
- Category

Output:

- Name
- Description

Sample Input

```
<soap:Envelope>
<soap:Header>
  <Authentication>
    <CustomerID>string</CustomerID>
    <UserName>string</UserName>
    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
<GetOilGasPoi xmlns="http://assetcontrolcenter.com/">
  <lat>double</lat>
  <lon>double</lon>
  <distanceFromCenter>double</distanceFromCenter>
  <name>string</name>
  <category>string</category>
</GetOilGasPoi>
</soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
<soap:Body>
  <GetOilGasPoiResponse xmlns="http://assetcontrolcenter.com/">
    <GetOilGasPoiResult>
      <PoiList>
        <OilGasPoiData>
          <Name>string</Name>
          <Description>string</Description>
        </OilGasPoiData>
        <OilGasPoiData>
          <Name>string</Name>
          <Description>string</Description>
        </OilGasPoiData>
      </PoiList>
    </GetOilGasPoiResult>
```

```
</GetOilGasPoiResponse>
</soap:Body>
</soap:Envelope>
```

Get Asset Odometer

Input:

- Unit number

Output:

- Odometer value
- Reading date (UTC time)

Sample Input

```
<soap:Envelope>
<soap:Header>
  <Authentication>
    <CustomerID>string</CustomerID>
    <UserName>string</UserName>
    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
<GetAssetOdometer xmlns="http://assetcontrolcenter.com/">
  <assetUnitNumber>string</assetUnitNumber>
</GetAssetOdometer>
</soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
<soap:Body>
<GetAssetOdometerResponse xmlns="http://assetcontrolcenter.com/">
  <GetAssetOdometerResult>
    <Odometer>int</Odometer>
    <ReadingDate>dateTime</ReadingDate>
    <UseECMOdometer>boolean</UseECMOdometer>
  </GetAssetOdometerResult>
</GetAssetOdometerResponse>
```

```
</soap:Body>
</soap:Envelope>
```

Get Asset Hour Meter

Input:

- Unit number

Output:

- Hour meter
- Reading date (UTC time)

Sample Input

```
<soap:Envelope>
<soap:Header>
  <Authentication>
    <CustomerID>string</CustomerID>
    <UserName>string</UserName>
    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
<GetAssetHourMeter xmlns="http://assetcontrolcenter.com/">
  <assetUnitNumber>string</assetUnitNumber>
</GetAssetHourMeter>
</soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
<soap:Body>
<GetAssetHourMeterResponse xmlns="http://assetcontrolcenter.com/">
  <GetAssetHourMeterResult>
    <HourMeter>double</HourMeter>
    <ReadingDate>dateTime</ReadingDate>
    <IsAssetTracker>boolean</IsAssetTracker>
    <UseECMHourMeter>boolean</UseECMHourMeter>
  </GetAssetHourMeterResult>
</GetAssetHourMeterResponse>
```

```
</soap:Body>
</soap:Envelope>
```

Reset Asset Odometer

Input:

- Unit number
- Odometer Value
- Effective Date (UTC time)

Output:

- True/False (True means success, False means failure)

Sample Input

```
<soap:Envelope>
<soap:Header>
  <Authentication>
    <CustomerID>string</CustomerID>
    <UserName>string</UserName>
    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
<ResetAssetOdometer xmlns="http://assetcontrolcenter.com/">
  <assetUnitNumber>string</assetUnitNumber>
  <odometerValue>int</odometerValue>
  <effectiveDate>dateTime</effectiveDate>
</ResetAssetOdometer>
</soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
<soap:Body>
<ResetAssetOdometerResponse xmlns="http://assetcontrolcenter.com/">
  <ResetAssetOdometerResult>boolean</ResetAssetOdometerResult>
</ResetAssetOdometerResponse>
</soap:Body>
</soap:Envelope>
```

Reset Asset Hour Meter

Input:

- Unit number
- Hour Meter
- Effective Date (UTC time)

Output:

- True/False (True means success, False means failure)

Sample Input

```
<soap:Envelope>
<soap:Header>
  <Authentication>
    <CustomerID>string</CustomerID>
    <UserName>string</UserName>
    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
<ResetAssetHourMeter xmlns="http://assetcontrolcenter.com/">
  <assetUnitNumber>string</assetUnitNumber>
  <hourMeterValue>double</hourMeterValue>
  <effectiveDate>dateTime</effectiveDate>
</ResetAssetHourMeter> </soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
<soap:Body>
  <ResetAssetHourMeterResponse xmlns="http://assetcontrolcenter.com/">
    <ResetAssetHourMeterResult>boolean</ResetAssetHourMeterResult>
  </ResetAssetHourMeterResponse>
</soap:Body>
</soap:Envelope>
```

Get ECM Odometer Reading

Input:

- Unit number

Output:

- Odometer Value
- Timestamp (UTC)
- MIM Protocol name
- GEOTrac MsgID

Sample Input

```
<soap:Envelope>
<soap:Header>
  <Authentication>
    <CustomerID>string</CustomerID>
    <UserName>string</UserName>
    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
<GetECMOdometerReading xmlns="http://assetcontrolcenter.com/">
  <assetUnitNumber>string</assetUnitNumber>
</GetECMOdometerReading>
</soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
<soap:Body>
<GetECMOdometerReadingResponse xmlns="http://assetcontrolcenter.com/">
  <GetECMOdometerReadingResult>
    <MimValue>double</MimValue>
    <MsgTime>dateTime</MsgTime>
    <MimProtocol>string</MimProtocol>
    <GEOTracMsgID>guid</GEOTracMsgID>
  </GetECMOdometerReadingResult>
</GetECMOdometerReadingResponse>
```

```
</soap:Body>  
</soap:Envelope>
```

Get ECM Hour Meter Reading

Input:

- Unit number

Output:

- Hour Meter
- Timestamp (UTC)
- MIM Protocol name
- GEOTrac MsgID

Sample Input

```
<soap:Envelope>
<soap:Header>
  <Authentication>
    <CustomerID>string</CustomerID>
    <UserName>string</UserName>
    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
<GetECMHourMeterReading xmlns="http://assetcontrolcenter.com/">
  <assetUnitNumber>string</assetUnitNumber>
</GetECMHourMeterReading>
</soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
<soap:Body>
  <GetECMHourMeterReadingResponse xmlns="http://assetcontrolcenter.com/">
    <GetECMHourMeterReadingResult>
      <MimValue>double</MimValue>
      <MsgTime>dateTime</MsgTime>
      <MimProtocol>string</MimProtocol>
      <GEOTracMsgID>guid</GEOTracMsgID>
    </GetECMHourMeterReadingResult>
  </GetECMHourMeterReadingResponse>
</soap:Body>
</soap:Envelope>
```


Get ECM Fuel Consumed Reading

Input:

- Unit number

Output:

- Fuel Consumed
- Timestamp (UTC)
- MIM Protocol name
- GEOTrac MsgID

Sample Input

```
<soap:Envelope>
<soap:Header>
  <Authentication>
    <CustomerID>string</CustomerID>
    <UserName>string</UserName>
    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
  <GetECMFuelConsumedReading xmlns="http://assetcontrolcenter.com/">
    <assetUnitNumber>string</assetUnitNumber>
  </GetECMFuelConsumedReading>
</soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
<soap:Body>
  <GetECMFuelConsumedReadingResponse xmlns="http://assetcontrolcenter.com/">
    <GetECMFuelConsumedReadingResult>
      <MimValue>double</MimValue>
      <MsgTime>dateTime</MsgTime>
      <MimProtocol>string</MimProtocol>
      <GEOTracMsgID>guid</GEOTracMsgID>
    </GetECMFuelConsumedReadingResult>
  </GetECMFuelConsumedReadingResponse>
</soap:Body>
</soap:Envelope>
```

IFTA Web Service

Purpose

The IFTA (International Fuel Tax Agreement) Web Service call will allow GEOTrac customers to consume their IFTA related information – specifically the time and distance driven in each particular IFTA compliant region - for the time period specified starting at 12:00:00 am on the start day specified and ending at 12:00:00 am on the end day specified for each unit . This web service using default timezone name string "(GMT-07:00) Mountain Time (US & Canada)".

Input

- Start day
- End day

Output:

- Unit number
- Modem serial number
- Group
- Day (single day from start end day)
- Province
- Distance driven (Km)
- Operation time (seconds)
- HH:MM:SS for operation time

Sample Input

```
<soap:Envelope>
  <soap:Header>
    <Authentication>
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
      <Password>string</Password>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <GetIFTA>
      <startDay>dateTime</startDay>
      <endDay>dateTime</endDay>
    </GetIFTA>
  </soap:Body>
```

</soap:Envelope>

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetIFTAResponse>
      <GetIFTAResult>
        <IFTAList>
          <IFTAData>
            <AssetUN>string</AssetUN>
            <ModemSN>string</ModemSN>
            <Group>string</Group>
            <Day>dateTime</Day>
            <Province>string</Province>
            <DistanceDriven>int</DistanceDriven>
            <OperationTime>int</OperationTime>
            <HhMmSs>string</HhMmSs>
          </IFTAData>
          <IFTAData>
            <AssetUN>string</AssetUN>
            <ModemSN>string</ModemSN>
            <Group>string</Group>
            <Day>dateTime</Day>
            <Province>string</Province>
            <DistanceDriven>int</DistanceDriven>
            <OperationTime>int</OperationTime>
            <HhMmSs>string</HhMmSs>
          </IFTAData>
        </IFTAList>
      </GetIFTAResult>
    </GetIFTAResponse>
  </soap:Body>
</soap:Envelope>
```

IFTA By Time zone Web Service

Purpose

The IFTA (International Fuel Tax Agreement) Web Service call will allow GEOTrac customers to consume their IFTA related information – specifically the time and distance driven in each particular IFTA compliant region - for the time period specified starting at 12:00:00 am on the start day specified and ending at 12:00:00 am on the end day specified for each unit . Given Timezone Name string format example is "(GMT-07:00) Mountain Time (US & Canada)" without quotation marks.

Input

- Start day
- End day
- Timezone Name

Output:

- Unit number
- Modem serial number
- Group
- Day (single day from start end day)
- Province
- Distance driven (Km)
- Operation time (seconds)
- HH:MM:SS for operation time

Sample Input

```
<soap:Envelope>  
  
<soap:Header>  
  
<Authentication xmlns="http://assetcontrolcenter.com/">  
  
<CustomerID>string</CustomerID>  
  
<UserName>string</UserName>  
  
<Password>string</Password>  
  
</Authentication>  
  
</soap:Header>  
  
<soap:Body>  
  
<GetIFTAByTimezone xmlns="http://assetcontrolcenter.com/">  
  
<startDay>dateTime</startDay>  
  
<endDay>dateTime</endDay>  
  
<timezoneName>string</timezoneName>  
  
</GetIFTAByTimezone>  
  
</soap:Body>  
  
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetIFTAResponse>
      <GetIFTAResult>
        <IFTAList>
          <IFTAData>
            <AssetUN>string</AssetUN>
            <ModemSN>string</ModemSN>
            <Group>string</Group>
            <Day>dateTime</Day>
            <Province>string</Province>
            <DistanceDriven>int</DistanceDriven>
            <OperationTime>int</OperationTime>
            <HhMmSs>string</HhMmSs>
          </IFTAData>
          <IFTAData>
            <AssetUN>string</AssetUN>
            <ModemSN>string</ModemSN>
            <Group>string</Group>
            <Day>dateTime</Day>
            <Province>string</Province>
            <DistanceDriven>int</DistanceDriven>
            <OperationTime>int</OperationTime>
            <HhMmSs>string</HhMmSs>
          </IFTAData>
        </IFTAList>
      </GetIFTAResult>
    </GetIFTAResponse>
  </soap:Body>
</soap:Envelope>
```

IFTA By Day Web Service

Purpose

The IFTA (International Fuel Tax Agreement) Web Service call will allow GEOTrac customers to consume their IFTA related information – specifically the time and distance driven in each particular IFTA compliant region - for the time period specified starting at 12:00:00 am on the start day specified and ending at 12:00:00 am on the end day specified for each unit . This web service using default timezone name string "(GMT-07:00) Mountain Time (US & Canada)".

Input

- Day

Output:

- Unit number
- Modem serial number
- Group
- Day (single day from start end day)
- Province
- Distance driven (Km)
- Operation time (seconds)
- HH:MM:SS for operation time

Sample Input

```
<soap:Envelope>
  <soap:Header>
    <Authentication>
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
      <Password>string</Password>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <GetIFTAByDay xmlns="http://assetcontrolcenter.com/">
      <day>dateTime</day>
    </GetIFTAByDay> </soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetIFTAByDayResponse xmlns="http://assetcontrolcenter.com/">
      <GetIFTAByDayResult>
        <IFTAList>
          <IFTAData>
            <AssetUN>string</AssetUN>
            <ModemSN>string</ModemSN>
            <Group>string</Group>
            <Day>dateTime</Day>
            <Province>string</Province>
            <DistanceDriven>int</DistanceDriven>
            <OperationTime>int</OperationTime>
            <HhMmSs>string</HhMmSs>
          </IFTAData>
          <IFTAData>
            <AssetUN>string</AssetUN>
            <ModemSN>string</ModemSN>
            <Group>string</Group>
            <Day>dateTime</Day>
            <Province>string</Province>
            <DistanceDriven>int</DistanceDriven>
            <OperationTime>int</OperationTime>
            <HhMmSs>string</HhMmSs>
          </IFTAData>
        </IFTAList>
      </GetIFTAByDayResult>
    </GetIFTAByDayResponse> </soap:Body>
</soap:Envelope>
```

IFTA By Day By Time zone Web Service

Purpose

The IFTA (International Fuel Tax Agreement) Web Service call will allow GEOTrac customers to consume their IFTA related information – specifically the time and distance driven in each particular IFTA compliant region - for the time period specified starting at 12:00:00 am on the start day specified and ending at 12:00:00 am on the end day specified for each unit . Given Timezone Name string format example is "(GMT-07:00) Mountain Time (US & Canada)" without quotation marks.

Input

- Day
- Timezone Name

Output:

- Unit number
- Modem serial number
- Group
- Day (single day from start end day)
- Province
- Distance driven (Km)
- Operation time (seconds)
- HH:MM:SS for operation time

Sample Input

```
<soap:Envelope >
```

```
<soap:Header>
```

```
<Authentication xmlns="http://assetcontrolcenter.com/">
```

```
<CustomerID>string</CustomerID>
```

```
<UserName>string</UserName>
```

```
<Password>string</Password>
```

```
</Authentication>
```

```
</soap:Header>
```

```
<soap:Body>
```

```
<GetIFTAByDayByTimezone xmlns="http://assetcontrolcenter.com/">
```

```
<day>dateTime</day>
```

```
<timezoneName>string</timezoneName>
```

```
</GetIFTAByDayByTimezone>
```

```
</soap:Body>
```

```
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetIFTAByDayByTimezoneResponse xmlns="http://assetcontrolcenter.com/">
      <GetIFTAByDayByTimezoneResult>
        <IFTAList>
          <IFTAData>
            <AssetUN>string</AssetUN>
            <ModemSN>string</ModemSN>
            <Group>string</Group>
            <Day>dateTime</Day>
            <Province>string</Province>
            <DistanceDriven>int</DistanceDriven>
            <OperationTime>int</OperationTime>
            <HhMmSs>string</HhMmSs>
          </IFTAData>
          <IFTAData>
            <AssetUN>string</AssetUN>
            <ModemSN>string</ModemSN>
            <Group>string</Group>
            <Day>dateTime</Day>
            <Province>string</Province>
            <DistanceDriven>int</DistanceDriven>
            <OperationTime>int</OperationTime>
            <HhMmSs>string</HhMmSs>
          </IFTAData>
        </IFTAList>
      </GetIFTAByDayByTimezoneResult>
    </GetIFTAByDayByTimezoneResponse>
  </soap:Body>
</soap:Envelope>
```

Engine Fault Code Web Service

Purpose

The Engine Fault Code Web Service call will allow GEOTrac customers to consume the actual Fault Codes that were generated, so customers can match them up to the manufacture's information that customers have to get the details on what actually occurred (such as Engine Coolant Level Sensor failed) for the specific message (for example, J1708 Fault Code).

NOTE: GEOTracMsgID is additional output in Activity Web Service. It is required input in Engine Fault Code Web Service.

Input

- GEOTracMsgID

Output:

- Dtc
- Value

Sample Input

```
<soap:Envelope>
  <soap:Header>
    <Authentication>
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
      <Password>string</Password>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <GetEngineFaultCode>
      <GEOTracMsgID>guid</GEOTracMsgID>
    </GetEngineFaultCode>
  </soap:Body>
</soap:Envelope>
```


Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetEngineFaultCodeResponse>
      <GetEngineFaultCodeResult>
        <EngineFaultCodeList>
          <EngineFaultCodeData>
            <Dtc>string</Dtc>
            <Value>int</Value>
          </EngineFaultCodeData>
          <EngineFaultCodeData>
            <Dtc>string</Dtc>
            <Value>int</Value>
          </EngineFaultCodeData>
        </EngineFaultCodeList>
      </GetEngineFaultCodeResult>
    </GetEngineFaultCodeResponse>
  </soap:Body>
</soap:Envelope>
```

Last Known Location

This web service call will provide the last known reported time and location of all customer vehicles, as well as the last reported state, heading and speed at the time the service is called. This service is designed to be called “occasionally” to get the last known state of all vehicles. If you are interested in knowing the near-real-time location then please refer to the “Activity Web Service”.

Input: none

Output (for each vehicle):

- Message time (DD/MM/YYYY hh:mm:ss)
- Modem serial number
- Unit number
- Reason description
- Latitude (degrees)
- Longitude(degrees)
- Heading (degrees)
- Speed (kmh)

Sample Input

```
<soap:Envelope>
  <soap:Header>
    <Authentication>
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
      <Password>string</Password>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <GetLastKnownLocation />
  </soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetLastKnownLocationResponse>
      <GetLastKnownLocationResult>
        <LastKnownLocationList>
          <LastKnownLocationData>
            <MsgTime>dateTime</MsgTime>
            <ModemSN>string</ModemSN>
            <AssetUN>string</AssetUN>
            <Reason>string</Reason>
            <RIndex>int</RIndex>
            <Latitude>double</Latitude>
            <Longitude>double</Longitude>
            <Heading>double</Heading>
            <Speed>int</Speed>
          </LastKnownLocationData>
          <LastKnownLocationData>
            <MsgTime>dateTime</MsgTime>
            <ModemSN>string</ModemSN>
            <AssetUN>string</AssetUN>
            <Reason>string</Reason>
            <RIndex>int</RIndex>
            <Latitude>double</Latitude>
            <Longitude>double</Longitude>
            <Heading>double</Heading>
            <Speed>int</Speed>
          </LastKnownLocationData>
        </LastKnownLocationList>
      </GetLastKnownLocationResult>
    </GetLastKnownLocationResponse>
  </soap:Body>
</soap:Envelope>
```

GEOFence

This web service call will retrieve a list of the companies, Active or Expired GEOFences. The response will include the radius for city, marker or port, but will be null for land descriptions and road GEOFence types.

Input: none

Output (for each GEOFence):

- GEOFence Name

- Type
- Lat
- Long
- Radius (null for Roadtype and Land Description)
- LSD (Legal Land Description)
- Description

Sample Input

```
<soap:Envelope>  
  <soap:Header>  
    <Authentication>  
      <CustomerID>string</CustomerID>  
      <UserName>string</UserName>  
      <Password>string</Password>  
    </Authentication>  
  </soap:Header>  
  <soap:Body>  
    <GetGEOFence/>  
  </soap:Body>  
</soap:Envelope>
```


Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetGEOFenceResponse>
      <GetGEOFenceResult>
        <GEOFenceList>
          <GEOFenceData>
            <GeoFenceName>string</GeoFenceName>
            <Type>int</Type>
            <Latitude>double</Latitude>
            <Longitude>double</Longitude>
            <Radius>int</Radius>
            <Lsd>string</Lsd>
            <Description>string</Description>
          </GEOFenceData>
          <GEOFenceData>
            <GeoFenceName>string</GeoFenceName>
            <Type>int</Type>
            <Latitude>double</Latitude>
            <Longitude>double</Longitude>
            <Radius>int</Radius>
            <Lsd>string</Lsd>
            <Description>string</Description>
          </GEOFenceData>
        </GEOFenceList>
      </GetGEOFenceResult>
    </GetGEOFenceResponse>
  </soap:Body>
</soap:Envelope>
```

Assets (GEOTrac GPS Equipment Assets)

This web service call will retrieve all registered GEOTrac equipment and associated information registered to the customer.

Input: none

Output (for each GEOTrac asset):

- AssetType
- CustomerGroup
- VIN
- AssetUN
- RegistrationNumber
- Country
- Province
- Make
- Model
- Year
- Driver
- InstallationDate (Day)
- CostCenter
- ModemSN
- EcmSN
- MdtSN
- GatewaySN
- DualModelmei
- UseECMHourMeter
- UseECMOdometer
- Odometer
- DateofOdometer (UTC time)
- HourMeter
- DataofHourMeter (UTC time)
- Description
- CabCardNumber
- PermitNumber
- BCGVW
- ABGVW
- SKGVW
- TireSize
- TruckMobile
- OffRoadConsumptionRate
- OffRoadIdleConsumptionRate
- FuelType

- EngineType
- LicensePlate
- SpeedLimit

Sample Input

```
<soap:Envelope>
  <soap:Header>
    <Authentication>
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
      <Password>string</Password>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <GetAssets/>
  </soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetAssetsResponse>
      <GetAssetsResult>
        <AssetList>
          <AssetData>
            <AssetType>string</AssetType>
            <CustomerGroup>string</CustomerGroup>
            <VIN>string</VIN>
            <AssetUN>string</AssetUN>
            <RegistrationNumber>string</RegistrationNumber>
            <Country>string</Country>
            <Province>string</Province>
            <Make>string</Make>
            <Model>string</Model>
            <Year>string</Year>
            <Driver>string</Driver>
            <CostCenter>string</CostCenter>
            <ModemSN>string</ModemSN>
            <InstallationDate xsi:nil="true" />
            <EcmSN>string</EcmSN>
            <DualModelmei>string</DualModelmei>
            <MdtSN>string</MdtSN>
            <GatewaySN>string</GatewaySN>
            <UseECMHourMeter>boolean</UseECMHourMeter>
            <UseECMOdometer>boolean</UseECMOdometer>
            <Odometer>int</Odometer>
            <DateofOdometer>dateTime</DateofOdometer>
            <HourMeter>double</HourMeter>
            <DataofHourMeter>dateTime</DataofHourMeter>
            <Description>string</Description>
```

```

<CabCardNumber>string</CabCardNumber>
<PermitNumber>string</PermitNumber>
<BCGVW>string</BCGVW>
<ABGVW>string</ABGVW>
<SKGVW>string</SKGVW>
<TireSize>string</TireSize>
<TruckMobile>string</TruckMobile>
<OffRoadConsumptionRate>double</OffRoadConsumptionRate>
<OffRoadIdleConsumptionRate>double</OffRoadIdleConsumptionRate>
<FuelType>string</FuelType>
<EngineType>string</EngineType>
<LicensePlate>string</LicensePlate>
<SpeedLimit>int</SpeedLimit>
</AssetData>
<AssetData>
<AssetType>string</AssetType>
<CustomerGroup>string</CustomerGroup>
<VIN>string</VIN>
<AssetUN>string</AssetUN>
<RegistrationNumber>string</RegistrationNumber>
<Country>string</Country>
<Province>string</Province>
<Make>string</Make>
<Model>string</Model>
<Year>string</Year>
<Driver>string</Driver>
<CostCenter>string</CostCenter>
<ModemSN>string</ModemSN>
<InstallationDate xsi:nil="true" />
<EcmSN>string</EcmSN>
<DualModelmei>string</DualModelmei>
<MdtSN>string</MdtSN>
<GatewaySN>string</GatewaySN>
<UseECMHourMeter>boolean</UseECMHourMeter>
<UseECMOdometer>boolean</UseECMOdometer>
<Odometer>int</Odometer>
<DateofOdometer>dateTime</DateofOdometer>
<HourMeter>double</HourMeter>
<DataofHourMeter>dateTime</DataofHourMeter>
<Description>string</Description>
<CabCardNumber>string</CabCardNumber>
<PermitNumber>string</PermitNumber>
<BCGVW>string</BCGVW>
<ABGVW>string</ABGVW>
<SKGVW>string</SKGVW>
<TireSize>string</TireSize>
<TruckMobile>string</TruckMobile>
<OffRoadConsumptionRate>double</OffRoadConsumptionRate>

```

```
<OffRoadIdleConsumptionRate>double</OffRoadIdleConsumptionRate>
<FuelType>string</FuelType>
<EngineType>string</EngineType>
<LicensePlate>string</LicensePlate>
<SpeedLimit>int</SpeedLimit>
</AssetData>
</AssetList>
</GetAssetsResult>
</GetAssetsResponse>
</soap:Body>
</soap:Envelope>
```

Messaging

The messaging web service calls will allow the potential to service a full messaging application. The SendTextMessage() call can be used to integrate a client application in to the GEOTrac messaging service and GEOTrac dispatch applications. The GetTextMessages() call can be used by a client to extract all messages returned from the truck for a particular customer.

High level features of GetTextMessage web service call:

- Given a subscriber Company Id, the client can retrieve a block of Text Messages from the GEOTrac Activity Message database for that Company Id
- The client will have the ability to request all messages from the Last Text Message Id requested. The message ID will be a 64bit integer that automatically increments for each message received.
- The number of text messages returned is limited to 500 per call.
- The Last Text Message Id will also be returned in the response to be kept for future calls.
- GEOTrac will keep a maximum of one calendar years' worth of text messages.
- Client will be able to make multiple requests for the same data records
- The recommended polling interval will be every 5 minutes.

Send Text Message

Input:

- Unit number
- Subject
- Body

Output: True\False per unit for success or failed. If failed, error message would be provided.

Sample Input

```
<soap:Envelope>
  <soap:Header>
    <Authentication>
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
      <Password>string</Password>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <SendTextMessage xmlns="http://assetcontrolcenter.com/">
      <assetUnitNumber>string</assetUnitNumber>
      <subject>string</subject>
      <body>string</body>
    </SendTextMessage>
  </soap:Body>
</soap:Envelope>
```

```
</soap:Body>  
</soap:Envelope>
```


Sample Response

```
<soap:Envelope>
  <soap:Body>
    <SendTextMessageResponse xmlns="http://assetcontrolcenter.com/">
      <SendTextMessageResult>
        <AssetUnitNumber>string</AssetUnitNumber>
        <IsSuccess>boolean</IsSuccess>
        <ErrorMessage>string</ErrorMessage>
      </SendTextMessageResult>
    </SendTextMessageResponse>
  </soap:Body>
</soap:Envelope>
```

Get Text Message

Input

- Last Text Message ID

Output (for each message):

- Message ID
- GEOTracMsgID
- Message Date Time (UTC time)
- Unit number
- Subject
- Body
- Size
- IsImportant
- ReceiverUserId
- ReceiverUserInfo
- Latitude
- Longitude

Sample Input

```
<soap:Envelope>
  <soap:Header>
    <Authentication>
      <CustomerID>string</CustomerID>
      <UserName>string</UserName>
      <Password>string</Password>
    </Authentication>
  </soap:Header>
  <soap:Body>
    <GetTextMessage>
      <LastMsgID>long</LastMsgID>
    </GetTextMessage>
  </soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetTextMessageResponse xmlns="http://assetcontrolcenter.com/">
      <GetTextMessageResult>
        <LastMsgID>long</LastMsgID>
        <Messages>
          <Inbox>
            <MsgId>long</MsgId>
            <GEOTracMsgID>guid</GEOTracMsgID>
            <AssetUN>string</AssetUN>
            <CustomerId>int</CustomerId>
            <Latitude>double</Latitude>
            <Longitude>double</Longitude>
            <Date>dateTime</Date>
            <IsImportant>boolean</IsImportant>
            <Subject>string</Subject>
            <Body>string</Body>
            <Size>short</Size>
            <ReceiverUserId>int</ReceiverUserId>
            <ReceiverUserInfo xsi:nil="true" />
          </Inbox>
          <Inbox>
            <MsgId>long</MsgId>
            <GEOTracMsgID>guid</GEOTracMsgID>
            <AssetUN>string</AssetUN>
            <CustomerId>int</CustomerId>
            <Latitude>double</Latitude>
            <Longitude>double</Longitude>
            <Date>dateTime</Date>
            <IsImportant>boolean</IsImportant>
            <Subject>string</Subject>
            <Body>string</Body>
            <Size>short</Size>
            <ReceiverUserId>int</ReceiverUserId>
            <ReceiverUserInfo xsi:nil="true" />
          </Inbox>
        </Messages>
      </GetTextMessageResult>
    </GetTextMessageResponse>
  </soap:Body>
```

</soap:Envelope>

Drivers

GetDriverList() web service call will allow a client to extract a list of all their GEOTrac registered drivers and associated contact, company, and driver-related information.

Input: none

Output:

- Employment number
- First name
- Middle name
- Last name
- Driver's license number
- License class
- License expiry
- Group
- Unit number
- Phone
- Mobile
- Email
- Position
- Address
- City
- Comments
- PSM ID
- DriverID Info List

Sample Input

```
<soap:Envelope>
  <Authentication xmlns="http://assetcontrolcenter.com/">
    <CustomerID>string</CustomerID>
    <UserName>string</UserName>
    <Password>string</Password>
  </Authentication>
</soap:Header>
<soap:Body>
  <GetDriverList xmlns="http://assetcontrolcenter.com/" />
</soap:Body>
</soap:Envelope>
```

Sample Response

```
<soap:Envelope>
  <soap:Body>
    <GetDriverListResponse xmlns="http://assetcontrolcenter.com/">
      <GetDriverListResult>
        <DriverList>
          <DriverData>
            <DriverKey>int</DriverKey>
            <EmploymentNumber>string</EmploymentNumber>
            <FirstName>string</FirstName>
            <LastName>string</LastName>
            <MiddleName>string</MiddleName>
            <DriverLicenseNumber>string</DriverLicenseNumber>
            <LicenseClass>string</LicenseClass>
            <LicenseExpiryDate>dateTime</LicenseExpiryDate>
            <UnitNumber>string</UnitNumber>
            <Phone>string</Phone>
            <Mobile>string</Mobile>
            <Email>string</Email>
            <Position>string</Position>
            <HiredDateString>string</HiredDateString>
            <DateLeftString>string</DateLeftString>
            <Address>string</Address>
            <City>string</City>
            <Comments>string</Comments>
            <PsmId>string</PsmId>
            <GroupName>string</GroupName>
            <DriverIdList xsi:nil="true" />
          </DriverData>
          <DriverData>
            <DriverKey>int</DriverKey>
            <EmploymentNumber>string</EmploymentNumber>
            <FirstName>string</FirstName>
            <LastName>string</LastName>
            <MiddleName>string</MiddleName>
            <DriverLicenseNumber>string</DriverLicenseNumber>
            <LicenseClass>string</LicenseClass>
            <LicenseExpiryDate>dateTime</LicenseExpiryDate>
            <UnitNumber>string</UnitNumber>
            <Phone>string</Phone>
            <Mobile>string</Mobile>
```

```
<Email>string</Email>
<Position>string</Position>
<HiredDateString>string</HiredDateString>
<DateLeftString>string</DateLeftString>
<Address>string</Address>
<City>string</City>
<Comments>string</Comments>
<PsmId>string</PsmId>
<GroupName>string</GroupName>
<DriverIdList xsi:nil="true" />
</DriverData>
</DriverList>
</GetDriverListResult>
</GetDriverListResponse>
</soap:Body>
</soap:Envelope>
```

URL to Service

URL to GEOTrac Web Service is

<http://www.assetcontrolcenter.com/GMWS/GEOTracWebService.asmx>