

# „Opis Interfejsu Wymiany Danych”

# Spis treści:

1	Wstęp.....	3
2	Schemat wymiany danych: .....	3
3	Opis wymagań dla transmitowanych danych.....	4
4	Metody.....	4
4.1	Logowanie.....	4
4.1.1	Klucz sesyjny .....	4
4.1.2	Funkcja: Login().....	4
4.1.3	Rozpoczęcie wymiany danych.....	4
4.2	GetVehicleList .....	5
4.3	GetVehicleListLastState .....	6
4.4	GetVehicleEvents .....	8
4.5	GetFuncModificationStatus .....	10
4.6	GetPhotoList .....	11
5.	Słowniki:.....	13
5.	WSDL.....	13

## 1 Wstęp

Niniejsza specyfikacja określa wymagania Zamawiającego dla zakresu danych otrzymywanych od operatorów dotyczących systemu komunalnego oraz sposobu ich wymiany między systemami informatycznymi Wykonawcy i Zamawiającego.

Jako zakres wymaganych informacji transmitowanych poprzez interfejs wymiany danych, należy przyjąć zakres danych zawartych w poszczególnych metodach szczegółowo opisanych w kolejnych rozdziałach.

System musi rejestrować w szczególności następujące zdarzenia:

Typ zdarzenia	Moment rejestracji*
Punkty jazdy	Gdy pojazd jest w ruchu nie rzadziej niż co 100 m lub co 30 sekund
Punkty postoju	Gdy pojazd stoi, nie rzadziej niż co 3 minuty.
Załadunek odpadów	Natychmiast po wystąpieniu zdarzenia
Wyładunek pojazdu	Natychmiast po wystąpieniu zdarzenia

- Wszystkie rejestrowane zdarzenia muszą posiadać identyfikator pojazdu, datę i czas oraz współrzędne geograficzne wyznaczone na podstawie systemu GPS.

## 2 Schemat wymiany danych:

Celem wymiany danych jest przekazanie do Systemu Zamawiającego, danych zarejestrowanych przez pojazdy Wykonawców. Wszelkie rejestrowane dane i opisane w niniejszym dokumencie informacje, powinny być na bieżąco (w trybie online) przekazywane w oparciu o interfejs wymiany danych wykonany w technologii usług internetowych. Transfer danych odbywa się za pomocą interfejsu wymiany danych opartym o usługę internetową udostępnioną przez Wykonawcę i działającym w oparciu o żądania HTTPS.

### 3 *Opis wymagań dla transmitowanych danych*

Transmisja danych z urządzeń o których mowa poniżej musi być jednoczesna z transmisją danych z systemu monitoringu GPS. Wszystkie zarejestrowane zdarzenia muszą być rozszerzone o dokładną datę i czas [zgodny z uniwersalnym czasem koordynowanym UTC(PL)] oraz współrzędne geograficzne zdarzeń wyznaczone na podstawie systemu GPS

## 4 *Metody*

### 4.1 *Logowanie*

#### 4.1.1 *Klucz sesyjny*

Celem zagwarantowania bezpieczeństwa na przesyłane przez wykonawców dane, wprowadzono autoryzację polegającą na wpisaniu nazwy użytkownika oraz hasła wraz z numerem operatora. Po uzupełnieniu danych pod warunkiem ich poprawności, zwracany jest tzw. klucz sesyjny, który jest niezbędny do wywołania poszczególnych funkcji (poza Login). Dzięki niemu uzyskiwany jest dostęp do danych Operatorów. W przypadku podania błędnego lub klucza który utracił ważność, wywoływana funkcja nie zwróci danych, lecz zasygnalizuje problem odpowiednim statusem.

Klucz sesyjny ma swój okres ważności. W przypadku, gdy przez pewien określony czas nie będzie wywołana żadna funkcja przy jego użyciu, zostanie uznany za nieważny.

#### 4.1.2 *Funkcja: Login()*

**Opis** Funkcja zwraca klucz sesyjny wykorzystywany do komunikacji z usługą Operatora.

**Wejście** Nazwa użytkownika, hasło oraz identyfikator operatora:

**User:** string,

**Pass:** string,

**CustomerNumber:** Integer.

**Wyjście** Klucz sesyjny jako ciąg znaków (string) pozwalający na wywołanie pozostałych funkcji.

#### 4.1.3 *Rozpoczęcie wymiany danych*

W celu ustalenia zakresu i ilości danych do pobrania z Systemu Operatora, System zamawiającego rozpoczyna wymianę danych od ustalenia momentu ostatniej zmiany danych w Systemie Operatora (w podziale na typy danych

i dokumentów). Jeżeli data ostatniej zmiany danych jest późniejsza, niż data ostatniego pobrania danych, system zamawiającego pobiera nowe dane, wykorzystując stosowne funkcje.

## 4.2 GetVehicleList

Funkcja zwraca listę pojazdów używanych przez Operatora.

### Wejście:

- Klucz sesyjny służący do uwierzytelnienia.
- ModifiedAfter: DateTime – data i czas od którego system powinien pobrać zmiany danych.

Nazwa funkcji:	GetVechicleList
Opis:	Lista pojazdów.
Wyjście:	Lista pojazdów w postaci dokumentu XML

Tabela 1. Opis parametrów metody GetVehicleList

Parametr:	Opis parametru:
id	Unikalny Identyfikator pojazdu
name	Nazwa
plate-number	Numer rejestracyjny
mark	Marka
model	Model
production-date	Data produkcji
kerb-weight	Masa własna
side-number	Numer boczny
last-modification	Data i czas ostatniej modyfikacji rekordu
disable-time	Data dezaktywacji pojazdu (w przypadku gdy pojazd nie jest już udostępniany)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="result">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="vehicle" maxOccurs="unbounded" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="name"
              maxOccurs="1" minOccurs="1" />
            <xs:element type="xs:string"
```



<b>long</b>	Długość geograficzna
<b>state-id</b>	Stan pojazdu (zgodnie ze słownikiem stanów pojazdów)
<b>direction</b>	Kierunek przemieszczania się pojazdu.
<b>velocity</b>	Prędkość pojazdu
<b>last-modification</b>	Data i czas ostatniej modyfikacji rekordu
<b>last-continous-data-event-modtime</b>	Wskazują datę i czas modyfikacji rekordu wskazanego przez pole last-continous-data-event-time
<b>last-continous-data-event-time</b>	Wskazuje zarejestrowaną datę i czas wystąpienia ostatniego zdarzenia z ciągu zdarzeń które nie zostaną już zmodyfikowane. Oznacza to, że wszystkie zdarzenia zarejestrowane do tego czasu zostały już przesłane, nie występują już uzupełnienia i modyfikacje danych. Wszystkie dodane i zmodyfikowane w przyszłości rekordy będą miały datę i czas większy lub równy wskazanemu w tym polu

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xs:element name="result">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="vehicle-state" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:unsignedInt" name="vehicle-id" maxOccurs="1" minOccurs="1">
                <xs:annotation>
                  <xs:documentation>Id pojazdu przesłane w funkcji GetVehicleList</xs:documentation>
                </xs:annotation></xs:element>
              <xs:element type="xs:dateTime" name="datetime" maxOccurs="1" minOccurs="1">
                <xs:annotation>
                  <xs:documentation>Data i czas danej</xs:documentation>
                </xs:annotation></xs:element>
              <xs:element type="xs:unsignedInt" name="state-id" maxOccurs="1" minOccurs="1"/>
              <xs:element type="xs:unsignedInt" name="velocity" maxOccurs="1" minOccurs="0"/>
              <xs:element name="gpscoordinates" maxOccurs="1" minOccurs="1">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:int" name="direction" maxOccurs="1" minOccurs="1" />
                    <xs:element type="xs:float" name="lat" maxOccurs="1" minOccurs="1"/>
                    <xs:element type="xs:float" name="long" maxOccurs="1" minOccurs="1"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element type="xs:dateTime" name="last-modification" maxOccurs="1" minOccurs="1"/>
  <xs:element type="xs:dateTime" name="last-continous-data-event-modtime" maxOccurs="1" minOccurs="1"/>
  <xs:element type="xs:dateTime" name="last-continous-data-event-time" maxOccurs="1" minOccurs="1"/>

```

```

</xs:sequence>
  <xs:attribute type="xs:string" name="id" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
  <xs:attribute type="xs:dateTime" name="update-date-time" use="required"/>
</xs:complexType>
</xs:element>
</xs:schema>

```

Rys. 2 Diagram XSD dla metody GetVehicleListLastState

#### 4.4 GetVehicleEvents

Funkcja zwraca dane dotyczące pojazdu za zadany zakres czasu.

<b>Opis:</b>	Dane dla pojazdu za zadany zakres czasu.
<b>Wyjście:</b>	XML opisujący zdarzenia wykonane w zadanym przedziale czasu

#### Wejście:

- Klucz sesyjny służący do uwierzytelnienia (SessionId)
- Identyfikator pojazdu (int, VehicleId)
- Data początkowa zakresu czasu dla danych(DateTime, DateTimeFrom)-opcjonalnie
- Data końcowa zakresu czasu dla danych(DateTime, DateTimeTo)-opcjonalnie
- ModifiedAfter: DateTime data i czas od którego pobrać dane-wymagane

Tabela 3. Opis parametrów metody GetVehicleEvents

Parametr:	Opis parametru:
<b>vehicle-id</b>	Unikalny identyfikator pojazdu
<b>id</b>	unikalny identyfikator zdarzenia
<b>lat</b>	Szerokość geograficzna
<b>long</b>	Długość geograficzna
<b>event-type-id</b>	Identyfikator typu zdarzenia (wg. Tabela 4)
<b>status-gps</b>	Namiar gps (zgodnie ze słownikiem statusów GPS)
<b>startdatetime</b>	Data i czas rozpoczęcia zdarzenia
<b>enddatetime</b>	Data i czas zakończenia zdarzenia
<b>gpscoordinates</b>	(Latitude, Longitude) Długość i szerokość geograficzna



<b>maxspeed</b>	Prędkość maksymalna od poprzedniego zarejestrowanego punktu w km/h (wymagane w przypadku jazdy)
<b>distance</b>	Dystans przebyty od ostatniego zdarzenia w metrach (wymagane w przypadku jazdy)
<b>direction</b>	Kierunek jazdy (azymut) w stopniach. (wymagane w przypadku jazdy)
<b>last-modification</b>	Data i czas ostatniej modyfikacji rekordu
<b>update-date-time</b>	Data i czas ostatniej aktualizacji danych

Tabela 4. Opis identyfikatorów zdarzeń zwracanych poprzez wywołanie metody GetVehicleEvents

Event-Type-Id	Zdarzenie
1	Jazda
2	Postój
3	Załadunek odpadów
4	Wyładunek odpadów

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="result">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="event" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:unsignedInt" name="vehicle-id" maxOccurs="1" minOccurs="1"/>
              <xs:element type="xs:unsignedInt" name="event-type-id" maxOccurs="1" minOccurs="1"/>
              <xs:element name="gpscoordinates" maxOccurs="1" minOccurs="1">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:float" name="lat" maxOccurs="1" minOccurs="1"/>
                    <xs:element type="xs:float" name="long" maxOccurs="1" minOccurs="1"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element type="xs:int" name="direction" maxOccurs="1" minOccurs="1" />
              <xs:element name="status-gps" type="xs:unsignedInt" maxOccurs="1" minOccurs="1">
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element type="xs:unsignedInt" name="maxspeed" maxOccurs="1" minOccurs="0"/>
          <xs:element type="xs:unsignedInt" name="distance" maxOccurs="1" minOccurs="0"/>
          <xs:element type="xs:dateTime" name="startdatetime" maxOccurs="1" minOccurs="1"/>
          <xs:element type="xs:dateTime" name="enddatetime" maxOccurs="1" minOccurs="1"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:element>
</xs:schema>
```

```

        <xs:element type="xs:dateTime" name="last-modification" maxOccurs="1" minOccurs="1"/>
    </xs:sequence>
    <xs:attribute type="xs:string" name="id" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute type="xs:dateTime" name="update-date-time" use="required"/>
</xs:complexType>
</xs:element>
</xs:schema>

```

Rys. 3 Diagram XSD dla metody GetVehicleEvents

## 4.5 GetFuncModificationStatus

<b>Nazwa funkcji:</b>	<b>GetFuncModificationStatus</b>
<b>Opis:</b>	Pobranie informacji o ostatniej aktualizacji danych dostępnych przez interfejsy do integracji
<b>Wyjście:</b>	Lista funkcji ze statusem modyfikacji w postaci dokumentu XML

### Wejście:

- Klucz sesyjny służący do uwierzytelnienia (SessionId)

Tabela 5. Opis parametrów zwracanych przez wywołanie metody GetFuncModificationStatus

Parametr:	Opis parametru:
name	Nazwa metody
last-modification	Data modyfikacji

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="result">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="functions">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="function" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="name"/>
                    <xs:element type="xs:dateTime" name="last-modification"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

```

    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Rys. 4 Diagram XSD dla metody GetFuncModificationStatus

## 4.6 GetPhotoList

Nazwa funkcji:	GetPhonoList
Opis:	Pobranie informacji o zarejestrowanych zdjęciach
Wyjście:	Lista opisów wykonanych zdjęć

### Wejście:

- Klucz sesyjny służący do uwierzytelnienia (SessionId)
- Indeks kontynuacji (ContinueIndex)
- Czas od którego należy pobrać dane (ModifiedAfter)

### Opis przesyłania danych dla funkcji z parametrem ContinueIndex

1. Klient wysyła żądanie przesłania danych zmodyfikowanych po zadanym czasie podając parametr ModifiedAfter. Parametru ContinueIndex nie przesyła (ustawia na wartość *null*);
2. Serwer przesyła dane zmodyfikowane po czasie podanym w parametrze ModifiedAfter. Jeśli serwer określi że do przesłania jest zbyt duża ilość danych, która z pewnych powodów (np. braku zasobów, zbyt długiego oczekiwania na odpowiedź) nie może zostać przesłana w jednym pakiecie może dane podzielić na mniejsze pakiety. W takim wypadku musi w dokumencie xml zawierającym zwracane dane ustawić argument continue-index określający punkt wznowienia
3. Klient odbiera dane. Ustawiony argument continue-index w dokumencie xml z danymi informuje go o tym że nie została przesłana cała porcja danych. Ponawia więc żądanie przesłania pozostałych danych ponownie podając parametr ModifiedAfter i parametr ContinueIndex ,którego wartość ustawia na tą przesłaną przez serwer.
4. Serwer przesyła dane zmodyfikowane po czasie podanym w parametrze ModifiedAfter i od punktu wznowienie przesłanego w parametrze continue-index. Jeśli pakiet danych jest nadal zbyt duży dzieli go i ustawia nowy continue-index.

5. Klient odbiera dane. Jeśli argument continue-index w dokumencie xml z danymi nie jest ustawiony uznaje że wszystkie dane dla zadanej funkcji zostały przesłane.

Tabela 6. Opis parametrów wejściowych

Parametr:	Opis parametru:
name	Nazwa metody
ModifiedAfter	DateTime data i czas od którego pobrać dane
ContinueIndex	Indeks kontynuacji

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="result">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ListPhoto" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="vehicle-id" type="xs:int" maxOccurs="1" minOccurs="1"/>
              <xs:element name="date" type="xs:string" maxOccurs="1" minOccurs="1"/>
              <xs:element name="camera" type="xs:int" maxOccurs="1" minOccurs="1"/>
              <xs:element name="gps-coordinates">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="lon" type="xs:float"/>
                    <xs:element name="lat" type="xs:float"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="url" type="xs:string" maxOccurs="1" minOccurs="1"/>
              <xs:element name="last-modification" type="xs:string" maxOccurs="1" minOccurs="1"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:attribute name="continue-index" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="last-modification" type="xs:string"></xs:attribute>
</xs:complexType>
</xs:element>
</xs:schema>
```

Rys. 4 Diagram XSD dla metody GetPhotoList

Tabela 7. Opis parametrów zwracanych przez wywołanie metody GetPhotoList

Parametr:	Opis parametru:
-----------	-----------------

vehicle-id	Identyfikator pojazdu
date	Data wykonania fotografii
camera	Numer kamery
lat	Szerokość geograficzna
lon	Długość geograficzna
url	Adres URL do zdjęcia
last-modification	Data i czas zmiany rekordu

## 5. Słowniki:

### 5.1 Słownik rodzajów namiarów GPS

status-gps	Nazwa
1	Namiar GPS zapewniony
2	Brak namiaru GPS

### 5.2 Słownik stanów pojazdów

state	Nazwa
1	Jazda
2	Postój

## 5. WSDL

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://localhost/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:ns1="urn:uXMLRemotable" xmlns:ns2="urn:" name="WebServiceService" targetNamespace="http://localhost/">
<types>
<xs:schema xmlns="urn:uXMLRemotable" targetNamespace="urn:uXMLRemotable">
<xs:complexType name="TXMLRemotable">
<xs:sequence>
<xs:element name="Status" type="xs:int"/>
<xs:element name="XML" type="xs:string"/>
</xs:sequence>
```

```

</xs:complexType>
</xs:schema>
</types>
<message name="LoginRequest">
  <part name="User" type="xs:string"/>
  <part name="Password" type="xs:string"/>
  <part name="CustomerNumber" type="xs:int"/>
</message>
<message name="LoginResponse">
  <part name="return" type="xs:string"/>
</message>
<message name="GetVehicleListRequest">
  <part name="SessionId" type="xs:string"/>
  <part name="ModifiedAfter" type="xs:dateTime"/>
</message>
<message name="GetVehicleListResponse">
  <part name="return" type="ns1:TXMLRemotable"/>
</message>
<message name="GetVehicleListLastStateRequest">
  <part name="SessionId" type="xs:string"/>
  <part name="ModifiedAfter" type="xs:dateTime"/>
</message>
<message name="GetVehicleListLastStateResponse">
  <part name="return" type="ns1:TXMLRemotable"/>
</message>
<message name="GetVehicleEventsRequest">
  <part name="SessionId" type="xs:string"/>
  <part name="VehicleId" type="xs:unsignedInt"/>
  <part name="ModifiedAfter" type="xs:dateTime"/>
  <part name="DateFrom" type="xs:dateTime"/>
  <part name="DateTo" type="xs:dateTime"/>
</message>
<message name="GetVehicleEventsResponse">
  <part name="return" type="ns1:TXMLRemotable"/>
</message>
<message name="GetFuncModificationStatusRequest">
  <part name="SessionId" type="xs:string"/>
</message>
<message name="GetFuncModificationStatusResponse">
  <part name="return" type="ns1:TXMLRemotable"/>
</message>
<message name="GetPhotoListRequest">
  <part name="SessionId" type="xs:string"/>
  <part name="ModifiedAfter" type="xs:dateTime"/>
  <part name="ContinueIndex" type="xs:string"/>
</message>
<message name="GetPhotoListResponse">
  <part name="return" type="ns1:TXMLRemotable"/>
</message>
<portType name="IWebService">
  <operation name="Login">
    <input message="tns:LoginRequest"/>
    <output message="tns:LoginResponse"/>
  </operation>
  <operation name="GetVehicleList">
    <input message="tns:GetVehicleListRequest"/>

```

```

    <output message="tns:GetVehicleListResponse"/>
  </operation>
  <operation name="GetVehicleListLastState">
    <input message="tns:GetVehicleListLastStateRequest"/>
    <output message="tns:GetVehicleListLastStateResponse"/>
  </operation>
  <operation name="GetVehicleEvents">
    <input message="tns:GetVehicleEventsRequest"/>
    <output message="tns:GetVehicleEventsResponse"/>
  </operation>
  <operation name="GetFuncModificationStatus">
    <input message="tns:GetFuncModificationStatusRequest"/>
    <output message="tns:GetFuncModificationStatusResponse"/>
  </operation>
    <operation name="GetPhotoList">
      <input message="tns:GetPhotoListRequest"/>
      <output message="tns:GetPhotoListResponse"/>
    </operation>
  </portType>
  <binding name="IWebServicebinding" type="tns:IWebService">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="Login">
      <soap:operation soapAction="urn:WebServiceIntf-IWebService#Login" style="rpc"/>
      <input>
        <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
      </input>
      <output>
        <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
      </output>
    </operation>
    <operation name="GetVehicleList">
      <soap:operation soapAction="urn:WebServiceIntf-IWebService#GetVehicleList" style="rpc"/>
      <input>
        <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
      </input>
      <output>
        <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
      </output>
    </operation>
    <operation name="GetVehicleListLastState">
      <soap:operation soapAction="urn:WebServiceIntf-IWebService#GetVehicleListLastState" style="rpc"/>
      <input>
        <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
      </input>
      <output>
        <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
      </output>
    </operation>
    <operation name="GetVehicleEvents">
      <soap:operation soapAction="urn:WebServiceIntf-IWebService#GetVehicleEvents" style="rpc"/>

```

```

<input>
  <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
</input>
<output>
  <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
</output>
</operation>
<operation name="GetFuncModificationStatus">
  <soap:operation soapAction="urn:WebServiceIntf-IWebService#GetFuncModificationStatus" style="rpc"/>
  <input>
    <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
  </input>
  <output>
    <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
  </output>
</operation>
  <operation name="GetPhotoList">
    <soap:operation soapAction="urn:WebServiceIntf-IWebService#GetPhotoList" style="rpc"/>
    <input>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
    </input>
    <output>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WebServiceIntf-IWebService"/>
    </output>
  </operation>
</binding>
<service name="IWebServiceservice">
  <port name="IWebServicePort" binding="tns:IWebServicebinding">
    <soap:address location="http://www.example.org"/>
  </port>
</service>
</definitions>

```