

OpenAPI server

General Description

Skånetrafikens Open API exposes functions for traffic- and journeyplanning information.

API functions offers online search for current information about:

- Stop Areas, their names and positions
- Address data in Scania, street names inc. house numbers and coordinates
- Skånetrafikens Point Of Interest data incl. their coordinates
- Journey searches from any type of point above as departure/destination point
- Searches for “stop areas near me” – stop areas in neighbourhood of one coordinate
- Searches for departure or arrival times on all stop areas
- Journey paths with all coordinates along particular journey
- Current traffic means, can be used to filter journey search to desired traffic means
- Current price information is included in results of each calculated journey
- Realtime information and traffic deviations messages are included in results from server, when they occurs on requested journeys or requested departure/arrival times on Skånetrafikens stop areas.

Server capacity

OpenAPI server is hosting a number of existing applications. Load tests from 08 maj 2014 showed that remaining, available server capacity is 10 simultaneous requests for calculations per second.

Communication and protocol

OpenAPI builds on request/response message exchange pattern in which a requestor sends a request message to a replier system which receives and processes the request, ultimately returning a message in response. pattern is implemented in a purely **synchronous** fashion as common in client-server architectures.

Communication protocol is HTTP, through the default port 80.
HTTP method used for requests to server is GET.

The server is open, does not require any authentication and authorisation from clients.

API Methods, Object Types and descriptions

Method name: Querypage

Request for search after departure or/and destination points.

Input parameters

inpPointFr free text search string, from point

inpPointTo free text search string, to point

Response schema

```
= <xs:element name="GetStartEndPointResult">
  = <xs:complexType>
    = <xs:sequence>
      = <xs:element name="Code" type="xs:int" minOccurs="1" maxOccurs="1">
        = <xs:annotation>
          <xs:documentation>Response code, 0 if successful operation</xs:documentation>
        </xs:annotation>
      </xs:element>
      = <xs:element name="Message" type="xs:string" minOccurs="1" maxOccurs="1">
        = <xs:annotation>
          <xs:documentation xml:lang="se">Error message, empty if response code = 0</
            xs:documentation>
        </xs:annotation>
      </xs:element>
      = <xs:element name="StartPoints" minOccurs="1" maxOccurs="1">
        = <xs:complexType>
          = <xs:sequence>
            <xs:element maxOccurs="unbounded" name="Point" type="Point" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      = <xs:element name="EndPoints" minOccurs="1" maxOccurs="1">
        = <xs:complexType>
          = <xs:sequence>
            <xs:element maxOccurs="unbounded" name="Point" type="Point" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      = <xs:element name="ViaPoints" minOccurs="0" maxOccurs="1">
        = <xs:annotation>
          <xs:documentation>[Optional] Element exists only if via point requested, the
            default is search without via point</xs:documentation>
        </xs:annotation>
        = <xs:complexType>
          = <xs:sequence>
            <xs:element maxOccurs="unbounded" name="Point" type="Point" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
= <xs:complexType name="Point">
  = <xs:sequence>
    = <xs:element name="Id" type="xs:int">
      = <xs:annotation>
        <xs:documentation>Unique Point (Stop Area, Address or Poi) Id</xs:documentation>
      </xs:annotation>
    </xs:element>
    = <xs:element name="Name" type="xs:string">
      = <xs:annotation>
        <xs:documentation>Stop Area, Address or Poi name as displayed for users</
          xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

```

    </xs:annotation>
  </xs:element>
  = <xs:element name="Type" type="PointType">
    = <xs:annotation>
      <xs:documentation>Denotes type of point (Stop Area, Address or Poi)</
        xs:documentation>
    </xs:annotation>
  </xs:element>
  = <xs:element name="X" type="xs:int">
    = <xs:annotation>
      <xs:documentation>X coordinate, RT90</xs:documentation>
    </xs:annotation>
  </xs:element>
  = <xs:element name="Y" type="xs:int">
    = <xs:annotation>
      <xs:documentation>Y coordinate, RT90</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:sequence>
</xs:complexType>

```

```

= <!--

```

Simple types in Response

```

-->
= <xs:simpleType name="PointType">
  = <xs:annotation>
    <xs:documentation>Denotes types of from / to points.</xs:documentation>
  </xs:annotation>
  = <xs:restriction base="xs:token">
    <xs:enumeration value="STOP_AREA" />
    <xs:enumeration value="ADDRESS" />
    <xs:enumeration value="POI" />
    <xs:enumeration value="UNKNOWN" />
  </xs:restriction>
</xs:simpleType>
= <!--

```

Method name: Resultspage

Request for search after journeys between two points obtained from querypage method.

Input parameters

inpTime [Optional] journey time, default is current time
inpDate [Optional] journey date , default is current date
selPointFr departure point name|id|type
selPointTo destination point name|id|type
NoOf [Optional] No of journeys in result, default is 5
transportMode [Optional] Sum of linetype ids retrieved from trafficmeans method
cmdAction search/next/previous set of journeys
LastStart [Optional] date&time of last journey in previous results (used in conjunction with cmdAction = next)
FirstStart [Optional] date&time of first journey in previous results (used in conjunction with cmdAction = previous)

```
= <xs:element name="GetJourneyResult">
  = <xs:complexType>
    = <xs:sequence>
      = <xs:element name="Code" type="xs:int" minOccurs="1" maxOccurs="1">
        = <xs:annotation>
          <xs:documentation>Response code, 0 if successful
            operation</xs:documentation>
        </xs:annotation>
      </xs:element>
      = <xs:element name="Message" type="xs:string" minOccurs="1"
        maxOccurs="1">
        = <xs:annotation>
          <xs:documentation xml:lang="se">Error message, empty if
            response code = 0</xs:documentation>
        </xs:annotation>
      </xs:element>
      = <xs:element name="JourneyResultKey" type="xs:string">
        = <xs:annotation>
          <xs:documentation>Used to uniquely identify resultset. This
            reference can be later passed to GetCachedResult
            function in order to retrieve previous resultset without
            making a new call to search engines</xs:documentation>
        </xs:annotation>
      </xs:element>
      = <xs:element minOccurs="1" maxOccurs="1" name="Journeys">
        = <xs:complexType>
          = <xs:sequence>
            = <xs:element minOccurs="0" maxOccurs="unbounded"
              name="Journey">
              = <xs:complexType>
                = <xs:sequence>
                  = <xs:element name="SequenceNo" type="xs:int" minOccurs="1"
                    maxOccurs="1">
                    = <xs:annotation>
                      <xs:documentation>Position in an ordered list of Journeys</
                    xs:documentation>
                    </xs:annotation>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          = <xs:element name="DepDateTime" type="xs:dateTime" minOccurs="1" maxOccurs="1">
            = <xs:annotation>
              <xs:documentation>Departure date and time.</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```

    = <xs:element name="ArrDateTime" type="xs:dateTime" minOccurs="1"
maxOccurs="1">
    = <xs:annotation>
    <xs:documentation>Arrival date and time.</xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="DepWalkDist"
    type="xs:int" minOccurs="1"
    maxOccurs="1">
    = <xs:annotation>
    <xs:documentation>Walk distance in m. between starting point for
journey (if address or POI) and Stop Area for departure</xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="ArrWalkDist" type="xs:int" minOccurs="1"
maxOccurs="1">
    = <xs:annotation>
    <xs:documentation>Walk distance in m. between Stop Area for arrival
and journey end point (if address or POI)</xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="NoOfChanges" type="xs:int" minOccurs="1"
maxOccurs="1">
    = <xs:annotation>
    <xs:documentation>No of changes</xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="Guaranteed" type="xs:boolean" minOccurs="1"
maxOccurs="1">
    = <xs:annotation>
    <xs:documentation>Denotes if journey is guaranteed by transport
authority, according to rules for "Travel Guarnantee"</xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="CO2factor" type="xs:int" minOccurs="1"
maxOccurs="1">
    = <xs:annotation>
    = <xs:documentation>
    Journeys impact on the environment - environmental index based
on the carbon dioxide (CO2) emissions, Values between 0(lowest impact) and 100
    = <xs:restriction base="xs:positiveInteger">
    <xs:minInclusive value="0" />
    <xs:maxInclusive value="100" />
    </xs:restriction>
    </xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="NoOfZones" type="xs:int" minOccurs="1"
maxOccurs="1">
    = <xs:annotation>
    <xs:documentation>No of passing zones in a zoned fare stucture
defined by transport authority</xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="PriceZoneList" type="xs:string">
    = <xs:annotation>
    <xs:documentation>Comma separated list of passing price zones</
xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="FareType" type="xs:string">
    = <xs:annotation>
    <xs:documentation>Current fare type</xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="Prices" minOccurs="1" maxOccurs="1">

```

```

    = <xs:complexType>
    = <xs:sequence>
    = <xs:element name="PriceInfo" minOccurs="0" maxOccurs="unbounded">
    = <xs:complexType>
    = <xs:sequence>
    = <xs:element name="PriceType" type="xs:string" minOccurs="1"
maxOccurs="1">
    = <xs:annotation>
    = <xs:documentation>Price type description (single fare, discount card
etc.)</xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="Price" type="xs:float" minOccurs="1" maxOccurs="1">
    = <xs:annotation>
    = <xs:documentation>Price for journey, in SEK</xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="VAT" type="xs:float" minOccurs="1"
maxOccurs="1">
    = <xs:annotation>
    = <xs:documentation>VAT, in SEK</xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="JourneyTicketKey" type="xs:string"
minOccurs="0" maxOccurs="1">
    = <xs:annotation>
    = <xs:documentation>Unique key used by the search
engine to identify an price object uniquely in the scope of different fare
types</xs:documentation>
    </xs:annotation>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    = <xs:element name="JourneyKey" type="xs:string"
minOccurs="1" maxOccurs="1">
    = <xs:annotation>
    = <xs:documentation>Used by the Elmer search engine to
identify an object uniquely in the scope of a traffic data. Information may be
used by back-end services like Map Service to draw itinerary on map.</
xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="RouteLinks" minOccurs="1"
maxOccurs="1">
    = <xs:complexType>
    = <xs:sequence>
    = <xs:element name="RouteLink" minOccurs="1"
maxOccurs="unbounded">
    = <xs:complexType>
    = <xs:sequence>
    = <xs:element name="RouteLinkKey" type="xs:string"
minOccurs="1" maxOccurs="1">
    = <xs:annotation>
    = <xs:documentation>Used by the Elmer search engine to
identify an object uniquely in the scope of a traffic data.</xs:documentation>
    </xs:annotation>
    </xs:element>
    = <xs:element name="DepDateTime" type="xs:dateTime"
minOccurs="1" maxOccurs="1">
    = <xs:annotation>
    = <xs:documentation>Departure date and time</
xs:documentation>

```

```

        </xs:annotation>
        </xs:element>
        = <xs:element name="DepIsTimingPoint" type="xs:boolean"
minOccurs="1" maxOccurs="1">
        = <xs:annotation>
        <xs:documentation>Denotes if Departure node is a timing
point. False means that DepDateTime is aproximated time</xs:documentation>
        </xs:annotation>
        </xs:element>
        = <xs:element name="ArrDateTime" type="xs:dateTime"
minOccurs="1" maxOccurs="1">
        = <xs:annotation>
        <xs:documentation>Arrival date and time</xs:documentation>
        </xs:annotation>
        </xs:element>
        = <xs:element name="ArrIsTimingPoint" type="xs:boolean"
minOccurs="1" maxOccurs="1">
        = <xs:annotation>
        <xs:documentation>Denotes if Arrival node is a timing point.
False means that ArrDateTime is aproximated time</xs:documentation>
        </xs:annotation>
        </xs:element>
        = <xs:element name="CallTrip" type="xs:boolean" minOccurs="1"
maxOccurs="1">
        = <xs:annotation>
        <xs:documentation>Denotes if vehicle run must be called in
advance ex. by phone</xs:documentation>
        </xs:annotation>
        </xs:element>
        = <xs:element name="PriceZones" minOccurs="1"
maxOccurs="1">
        = <xs:complexType>
        = <xs:sequence>
        = <xs:element name="PriceZone" minOccurs="0"
maxOccurs="unbounded">
        = <xs:complexType>
        = <xs:sequence>
        = <xs:element name="Id" type="xs:int" minOccurs="1"
maxOccurs="1">
        = <xs:annotation>
        <xs:documentation>Price zone's id</xs:documentation>
        </xs:annotation>
        </xs:element>
        </xs:sequence>
        </xs:complexType>
        </xs:element>
        </xs:sequence>
        </xs:complexType>
        </xs:element>
        = <xs:element name="RealTime" minOccurs="1" maxOccurs="1">
        = <xs:complexType>
        = <xs:sequence>
        = <xs:element name="RealTimeInfo" minOccurs="0"
maxOccurs="unbounded">
        = <xs:complexType>
        = <xs:sequence>
        = <xs:element name="NewDepPoint" type="xs:string"
minOccurs="0" maxOccurs="1">
        = <xs:annotation>
        <xs:documentation>Stop Point deviation from timetable (on
departure side). Information about new Stop Point</xs:documentation>
        </xs:annotation>
        </xs:element>
        = <xs:element name="NewArrPoint" type="xs:string" minOccurs="0" maxOccurs="1">
= <xs:annotation>

```

```

<xs:documentation>Stop Point deviation from timetable (on arrival side). Information about new Stop Point</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="DepTimeDeviation" type="xs:int" minOccurs="0"
maxOccurs="1">
  _ <xs:annotation>
<xs:documentation>Deviation from timetable time in min. (on departure side). Delays are positive integer values and earlier times are negative.</
xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="DepDeviationAffect" type="RealTimeAffect"
minOccurs="0" maxOccurs="1">
<xs:annotation>

<xs:documentation>Describes how departure time deviation affects the journey.</
xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="ArrTimeDeviation" type="xs:int" minOccurs="0" maxOccurs="1">
<xs:annotation><xs:documentation>Deviation from timetable time in min. (on arrival side). Delays are positive integer values and earlier times are negative.</
xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="ArrDeviationAffect" type="RealTimeAffect" minOccurs="0"
maxOccurs="1">
<xs:annotation>
<xs:documentation>Describes how arrival time deviation affects the journey.</
xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Canceled" type="xs:boolean" minOccurs="0" maxOccurs="1">
<xs:annotation>
<xs:documentation>Denotes if vehicle run is canceled. This event can impact the whole journey</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="RealTimeRoadInfo" minOccurs="0" maxOccurs="unbounded">
<xs:annotation>
<xs:documentation>RoadInfo can be present only if journey is a car route.</
xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element name="Road" type="xs:string" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Road number</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Reason" type="RoadInfoType" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Type of event</xs:documentation>
</xs:annotation></xs:element>
<xs:element name="Info" type="xs:string" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Info text</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```



```

</xs:element>
<xs:element name="From" minOccurs="1" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:element name="Id" type="xs:int" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Unique Stop area ID</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Name" type="xs:string" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Stop area name</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="StopPoint" type="xs:string" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Stop point name within the Stop Area</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="To" minOccurs="1" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:element name="Id" type="xs:int" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Unique Stop area ID</xs:documentation> </xs:annotation>
</xs:element>
<xs:element name="Name" type="xs:string" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Stop area name</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="StopPoint" type="xs:string" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Stop point name</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Line" minOccurs="1" maxOccurs="1">
<xs:complexType>
<xs:sequence>
<xs:element name="Name" type="xs:string" minOccurs="1" maxOccurs="1">
<xs:annotation><xs:documentation>Line's name</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="No" type="xs:int" minOccurs="0" maxOccurs="1">
<xs:annotation>
<xs:documentation>Line's number</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="RunNo" type="xs:int" minOccurs="0" maxOccurs="1">
<xs:annotation>
<xs:documentation>Line's run number</xs:documentation>
</xs:annotation>
</xs:element>
...
= <xs:element name="Distance" type="xs:int" minOccurs="1" maxOccurs="1">
=   <xs:annotation>
=     <xs:documentation>Distance i meters.</xs:documentation>
=   </xs:annotation>
</xs:element>
= <xs:element name="CO2value" type="xs:float" minOccurs="1" maxOccurs="1">
=   <xs:annotation>

```

```

        <xs:documentation>CO2 value in kg/person/km</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

```

<!--

```

Simple types in Response

```

-->
<xs:simpleType name="RoadInfoType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="UNKNOWN" />
    <xs:enumeration value="WEATHER" />
    <xs:enumeration value="ROAD_ACCIDENT" />
    <xs:enumeration value="ROAD_WORK" />
    <xs:enumeration value="ROAD_CONDITION" />
    <xs:enumeration value="EVENT" />
    <xs:enumeration value="HOLIDAY" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="RealTimeAffect">
  <xs:restriction base="xs:token">
    <xs:enumeration value="CRITICAL" />
    <xs:enumeration value="NON_CRITICAL" />
    <xs:enumeration value="PASSED" />
    <xs:enumeration value="NONE" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ScopeAttributeType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="CONCERNS_DEPARTURE" />
    <xs:enumeration value="CONCERNS_ARRIVAL" />
    <xs:enumeration value="CONCERNS_LINE" />
    <xs:enumeration value="CONCERNS_DEPARR" />
  </xs:restriction>
</xs:simpleType>

```

Method name: Stationresults

Request for search after departure or arrival times from stop area.

Input parameters

selPointFrKey	Stop Area id
inpDate	[Optional] date for departure
inpTime	[Optional] time for departure
selDirection	Optional] 0 = departures, 1 = arrivals

Response schema

```
+ <xs:element name="GetDepartureArrivalResult">
  - <xs:simpleType name="RoadInfoType">
    - <xs:restriction base="xs:token">
      <xs:enumeration value="UNKNOWN" />
      <xs:enumeration value="WEATHER" />
      <xs:enumeration value="ROAD_ACCIDENT" />
      <xs:enumeration value="ROAD_WORK" />
      <xs:enumeration value="ROAD_CONDITION" />
      <xs:enumeration value="EVENT" />
      <xs:enumeration value="HOLIDAY" />
    </xs:restriction>
  </xs:simpleType>
  - <xs:simpleType name="RealTimeAffect">
    - <xs:restriction base="xs:token">
      <xs:enumeration value="CRITICAL" />
      <xs:enumeration value="NON_CRITICAL" />
      <xs:enumeration value="PASSED" />
      <xs:enumeration value="NONE" />
    </xs:restriction>
  </xs:simpleType>
  - <xs:simpleType name="ScopeAttributeType">
    - <xs:restriction base="xs:token">
      <xs:enumeration value="CONCERNS_DEPARTURE" />
      <xs:enumeration value="CONCERNS_ARRIVAL" />
      <xs:enumeration value="CONCERNS_LINE" />
      <xs:enumeration value="CONCERNS_DEPARR" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Method name: Neareststation

Request for search after stop areas near by one coordinate.

Input parameters

X x coordinate, RT 90 system

Y y coordinate, RT 90 system

R radius in meters

Response schema

± <xs:element name="GetNearestStopAreaResult">

Method name: JourneyPath

Request for calculation of coordinates along particular journey retrieved in response from Resultspage.

Input parameters

cf JourneyKey retrieved in response from resultspage method

id SequenceNo of journey retrieved in response from resultspage method

Response schema

```
= <xs:element name="GetJourneyPathResult">
  = <xs:complexType>
    = <xs:sequence>
      = <xs:element name="Code" type="xs:int" minOccurs="1" maxOccurs="1">
        = <xs:annotation>
          <xs:documentation>Response code, 0 if successful
            operation</xs:documentation>
        </xs:annotation>
      </xs:element>
      = <xs:element name="Message" type="xs:string" minOccurs="1"
        maxOccurs="1">
        = <xs:annotation>
          <xs:documentation xml:lang="se">Error message, empty if
            response code = 0</xs:documentation>
        </xs:annotation>
      </xs:element>
      = <xs:element name="ResultXML" nillable="false" minOccurs="1"
        maxOccurs="1">
        = <xs:annotation>
          <xs:documentation>XML string</xs:documentation>
        </xs:annotation>
      - <!--
        Unescaped XML data
      -->
      = <xs:complexType>
        = <xs:sequence>
          = <xs:element name="Part">
            = <xs:complexType>
              = <xs:sequence>
                = <xs:element name="From">
                  = <xs:complexType>
                    = <xs:sequence>
                      <xs:element name="Id" type="xs:unsignedInt" />
                      <xs:element name="Poi" type="xs:unsignedByte" />
                      <xs:element name="PoiAlias" type="xs:unsignedByte" />
                      <xs:element name="Name" type="xs:string" />
                      <xs:element name="X" type="xs:decimal" />
                      <xs:element name="Y" type="xs:decimal" />
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                = <xs:element name="To">
                  = <xs:complexType>
                    = <xs:sequence>
                      <xs:element name="Id" type="xs:unsignedInt" />
                      <xs:element name="Poi" type="xs:unsignedByte" />
                      <xs:element name="PoiAlias" type="xs:unsignedByte" />
                      <xs:element name="Name" type="xs:string" />
                      <xs:element name="X" type="xs:decimal" />
                      <xs:element name="Y" type="xs:decimal" />
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:sequence>
  </xs:element>
```

```

= <xs:element name="Line">
  = <xs:complexType>
    = <xs:sequence>
      <xs:element name="Name" type="xs:string" />
      <xs:element name="No" type="xs:unsignedShort" />
      <xs:element name="LinTName" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
= <xs:element name="Coords">
  = <xs:complexType>
    = <xs:sequence>
      = <xs:element maxOccurs="unbounded" name="Coord">
        = <xs:complexType>
          = <xs:sequence>
            <xs:element name="X" type="xs:decimal" />
            <xs:element name="Y" type="xs:decimal" />
          </xs:sequence>
        </
          xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

Method name: Trafficmeans

Request for search after current trafficmeans.

Input parameters

Response schema

```
= <xs:element name="GetMeansOfTransportResult">
  = <xs:complexType>
    = <xs:sequence>
      = <xs:element name="Code" type="xs:int" minOccurs="1" maxOccurs="1">
        = <xs:annotation>
          <xs:documentation>Response code, 0 if successful operation</xs:documentation>
        </xs:annotation>
      </xs:element>
      = <xs:element name="Message" type="xs:string" minOccurs="1" maxOccurs="1">
        = <xs:annotation>
          <xs:documentation xml:lang="se">Error message, empty if response code = 0</
            xs:documentation>
        </xs:annotation>
      </xs:element>
      = <xs:element name="TransportModes" minOccurs="1" maxOccurs="1">
        = <xs:complexType>
          = <xs:sequence>
            = <xs:element maxOccurs="unbounded" name="TransportMode">
              = <xs:complexType>
                = <xs:sequence>
                  = <xs:element name="Id" type="xs:int" minOccurs="1" maxOccurs="1">
                    = <xs:annotation>
                      <xs:documentation>Unique Transport Mode ID</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  = <xs:element name="Name" type="xs:string" minOccurs="1"
                    maxOccurs="1">
                    = <xs:annotation>
                      <xs:documentation>Transtport mode name i.e. bus, train, ferry
                        etc.</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  = <xs:element name="DefaultChecked" type="xs:boolean"
                    minOccurs="1" maxOccurs="1">
                    = <xs:annotation>
                      <xs:documentation>Denotes if item should be selected by
                        default. Selection is made by transport authority</
                          xs:documentation>
                    </xs:annotation>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
= <xs:element name="LineTypes" minOccurs="1" maxOccurs="1">
  = <xs:complexType>
    = <xs:sequence>
      = <xs:element maxOccurs="unbounded"
        name="LineType">
        = <xs:complexType>
          = <xs:sequence>
            = <xs:element name="Id" type="xs:int" minOccurs="1" maxOccurs="1">
              = <xs:annotation>
```

```

        <xs:documentation>Unique line type ID</xs:documentation>
    </xs:annotation>
</xs:element>
= <xs:element name="Name" type="xs:string" minOccurs="1"
    maxOccurs="1">
    = <xs:annotation>
        <xs:documentation>Line type name</xs:documentation>
    </xs:annotation>
</xs:element>
= <xs:element name="DefaultChecked" type="xs:boolean" minOccurs="1"
    maxOccurs="1">
    = <xs:annotation>
        <xs:documentation>Denotes if item should be selected by default.
            Selection is made by transport authority</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```