TfN Open Data Hub API specification

Author: Martin Gogov, David Mountain

Date: 5 May 2020

Revision history: v0.1.8

transportapi

Table of Contents

1 Overview

This is an overview of the Transport for the North suite of APIs delivered by TransportAPI.

1.1 SIRI SX foundations

The API follows the SIRI SX standard as per its official definition (relevant parts below):

- Part 1: Context and framework (CEN/TS 15531-1:2015)
- Part 2: Communications infrastructure (CEN/TS 15531-2:2015)
- Part 5: Functional service interfaces – Situation Exchange (CEN/TS 15531-05:2016)

1.2 Transport layer

The transport layer used in this implementation is webhooks or "HTTP POST" as per SIRI SX: Part 1, section 3.2.70.

1.3 API credentials

Accounts and applications are created via the 3scale Open Data Hub available at https://opendata.transportforthenorth.com/.
API credentials can be found within the "Applications" tab of your developer account on the Open Data Hub portal if you are the Account admin.
For other users, your Account admin will be responsible for inviting users to your organisational Account.
You will need to accept your email invite and register to gain access to the Open Data Hub portal. Once logged in you will be able to access the Applications tab of your developer account to locate the credentials you need.

1.4 Authentication and authorization

Authorization is done by using the standard HTTP `Authorization` header with the `app_id` and `app_key` of the Open Data Hub 3scale application that should gatekeep the access to the API.
For example, if the 3scale application's `app_id=foo`, `app_key=bar`, then the `Authorization` header should contain the Base64 encoded form of the `app_id` / `app_key` value pair, separated by a colon `:`, i.e. the Base64 encoded form of `foo:bar` i.e. `Zm9vOmJhcgo=`, additionally prepended by `Basic` which means use HTTP Basic authentication i.e. the whole header would look like:

```
Authorization: Basic Zm9vOmJhcgo=
```

1.5 Common elements

All XML below would have the standard `<?xml>` element at the beginning but it will be omitted in this document for brevity:

```
<?xml version="1.0" encoding="UTF-8"?>
```

All HTTP requests below would have these common traits (the `<Siri>` XM element would be abbreviated to just `<Siri>` without any of its XML attributes in this document for brevity)

```
POST /siri/sx HTTP/1.1
Host: api.transportforthenorth.com
Content-Type: application/xml

Authorization: Basic Zm9vOmJhcgo=

<Siri
  xmlns="http://www.siri.org.uk/siri"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="2.0"
  xsi:schemaLocation="http://www.siri.org.uk/siri
http://www.siri.org.uk/schema/2.0/xsd/siri.xsd"
>
```

2 Polling API reference

The polling section of the API follows the Request/Response pattern as defined in `SIRI: Part 2, section 5.1.2 Request/Response Pattern.`

2.1 <ServiceRequest> request

```
<Siri>
  <!--
    * Count: exactly 1.
    Request from a for immediate delivery of data.
  -->
  <ServiceRequest>
    <!--
      * Count: exactly 1.
      Timestamp of generating the request specified by the client.
    -->
    <RequestTimestamp>2020-02-18T15:37:06+0000</RequestTimestamp>

    <!--
      * Count: exactly 1.
      The app_id of the Open Data Hub 3scale application should be put here.
```

```
    In case it is not valid or different to the app_id used in the HTTP Authorize
    header, an error would be returned.
  -->
  <RequestorRef>d0ab77fe</RequestorRef>

  <!--
    * Count: 0 or 1.
    An optional unique identifier of the service request assigned by the client.
    If provided, the same value is used to populate a corresponding
    <RequestMessageRef> element in the <ServiceDelivery> element in the response
    for debug and audit purposes.
  -->
  <MessageIdentifier>abcdef123456</MessageIdentifier>

  <!--
    * Count: 0, 1 or many.
    Request for information about the active transport situations from the
    Situation Exchange (SX) service.
   -->
  <SituationExchangeRequest version="2.0">
    <!--
      * Count: exactly 1.
      Timestamp of generating the request specified by the client.
    -->
    <RequestTimestamp>2020-02-18T15:37:06+0000</RequestTimestamp>

    <!--
      * Count: 0 or 1.
      Filters down to only the Situations which have some overlap of their
      validity periods, with the specified validity period below i.e.
      any of the <PtSituationElement> -> <ValidityPeriod> time intervals
      overlaps with the time interval specified in the <ValidityPeriod> below.
      Doesn't take publication windows, repetitions and day types into account.
    -->
    <ValidityPeriod>
      <!--
        * Count: exactly 1.
        Start timestamp of the validity period.
      -->
      <StartTime>2020-02-20T12:00:00+0000</StartTime>

      <!--
        * Count: 0 or 1.
        The end timestamp. If omitted, the range end is open-ended
        that is, it should be interpreted as "forever".
      -->
```

```
    <EndTime>2020-02-28T12:00:00+0000</EndTime>
  </ValidityPeriod>

  <!--
    * Count: 0 or 1.
    An element used to filter down situation based on transport network elements.
  -->
  <SituationNetworkFilter>
    <!--
      * Count: 0 or 1.
      If present, filter down to only the situations which affect the
      specified operator ref i.e. within the corresponding <PtSituationElement>
      element there is at least one
      <Consequences> -> <Consequence> -> <Affects> -> <Networks> ->
      <AffectedNetwork> -> <AffectedLine> -> <AffectedOperator> -> <OperatorRef>
      element with value matching the given operator ref.
      If missing, show situations for all operators.
    -->
    <OperatorRef>FLDS</OperatorRef>

    <!--
      * Count: 0 or 1.
      If present, filter down to only the situations which affect
      any of the specified stop point refs. Logically ORed i.e. within the
      corresponding <PtSituationElement> element there is at least one
      <Consequences> -> <Consequence> -> <Affects> -> <StopPoints> ->
      <AffectedStopPoint> -> <StopPointRef> element with value matching any of
      the given stop point refs.
      If missing, show situations for all stops.
      Additionally, filtering by bus stop prefix is also supported via e.g.
      filtering for 450* means filter down to bus stops with stop point refs
      which have the 450 prefix.
    -->
    <StopPointRef>450010197</StopPointRef>
    <StopPointRef>450010198</StopPointRef>
  </SituationNetworkFilter>

  <!--
    * Count: 0 or 1.
    An element used to filter down situation based on transport place elements.
  -->
  <SituationPlaceFilter>
    <!--
      * Count: 0 or 2.
      When 2, the two locations should specifying any two opposite corners
      of a bounding box.
```

```
          Filter down to only the situations which at least one affected stop
          point location which fall within the specified bounding box i.e.
          within the corresponding <PtSituationElement> element there is at least one
          <Consequences> -> <Consequence> -> <Affects> -> <StopPoints> ->
          <AffectedStopPoint> -> <Location> element which
          describes a location within the specified bounding box below.
        -->
        <Location srsName="wgs84">
          <!--
            * Count: 1.
            Specifies the longitude of the location.
          -->
          <Longitude>-1.54470907931</Longitude>

          <!--
            * Count: 1.
            Specifies the latitude of the location.
          -->
          <Latitude>53.80047655741</Latitude>
        </Location>
      </SituationPlaceFilter>
    </SituationExchangeRequest>
  </ServiceRequest>
</Siri>
```

Example 1: request all situations without any filters

```
<Siri>
  <ServiceRequest>
    <RequestTimestamp>2020-02-18T15:37:06+0000</RequestTimestamp>
    <RequestorRef>d0ab77fe</RequestorRef>

    <SituationExchangeRequest version="2.0">
      <RequestTimestamp>2020-02-18T15:37:06+0000</RequestTimestamp>
    </SituationExchangeRequest>
  </ServiceRequest>
</Siri>
```

Example 2: request all situations which relate to operator "FLDS" AND are valid for some time during the Feb 22, 23 weekend of 2020 AND affect any of the stops with ATCO codes 450010197, 450010198 or 450010199 AND have at least one affected stop with a location within the bounding box defined by the two points (53.80047655741, -1.54470907931) and (53.75047655741, -1.57470907931).

```
<Siri>
  <ServiceRequest>
    <RequestTimestamp>2020-02-18T12:30:00+0000</RequestTimestamp>
    <RequestorRef>d0ab77fe</RequestorRef>
```

```
    <SituationExchangeRequest version="2.0">
      <RequestTimestamp>2020-02-18T12:30:00+0000</RequestTimestamp>

      <ValidityPeriod>
        <StartTime>2020-02-22T00:00:00+0000</StartTime>
        <EndTime>2020-02-23T23:59:59+0000</EndTime>
      </ValidityPeriod>

      <SituationNetworkFilter>
        <OperatorRef>FLDS</OperatorRef>
        <StopPointRef>450010197</StopPointRef>
        <StopPointRef>450010198</StopPointRef>
        <StopPointRef>450010199</StopPointRef>
      </SituationNetworkFilter>

      <SituationPlaceFilter>
        <Location srsName="wgs84">
          <Longitude>-1.54470907931</Longitude>
          <Latitude>53.80047655741</Latitude>
        </Location>

        <Location srsName="wgs84">
          <Longitude>-1.57470907931</Longitude>
          <Latitude>53.75047655741</Latitude>
        </Location>
      </SituationPlaceFilter>
    </SituationExchangeRequest>
  </ServiceRequest>
</Siri>
```

Example 3: Request all situations which relate to operators FLDS or FWYO.

```
<Siri>
  <ServiceRequest>
    <RequestTimestamp>2020-02-18T12:30:00+0000</RequestTimestamp>
    <RequestorRef>d0ab77fe</RequestorRef>

    <SituationExchangeRequest version="2.0">
      <RequestTimestamp>2020-02-18T12:30:00+0000</RequestTimestamp>

      <SituationNetworkFilter>
        <OperatorRef>FLDS</OperatorRef>
      </SituationNetworkFilter>
    </SituationExchangeRequest>

    <SituationExchangeRequest version="2.0">
      <RequestTimestamp>2020-02-18T12:30:00+0000</RequestTimestamp>
```

```
      <SituationNetworkFilter>
        <OperatorRef>FWYO</OperatorRef>
      </SituationNetworkFilter>
    </SituationExchangeRequest>
  </ServiceRequest>
</Siri>
```

2.2 <ServiceDelivery> response

```
<Siri>
  <!--
    * Count: exactly 1.
    An element delivering payload data as a response to a <ServiceRequest>.
   -->
  <ServiceDelivery>
    <!--
      * Count: exactly 1.
      Timestamp of generation of the response by the server.
    -->
    <ResponseTimestamp>2020-02-18T12:30:30+0000</ResponseTimestamp>

    <!--
      * Count: exactly 1.
      Producer ref corresponding to the vendor.
     -->
    <ProducerRef>TransportAPI</ProducerRef>

    <!--
      * Count: exactly 1.
      Unique server-generated identifier of the response for audit and
      investigation purposes.
    -->
    <ResponseMessageIdentifier>abcdef123456</ResponseMessageIdentifier>

    <!--
      * Count: 0 or 1.
      Mirrors the <ServiceRequest> -> <MessageIdentifier> value if such was
      specified or else not present.
    -->
    <RequestMessageRef>qwerty654321</RequestMessageRef>

    <!--
      * Count: 0, 1 or many.
      An element delivering payload data as a response to a
```

```
      <SituationExchangeRequest>. One <SituationExchangeDelivery> element
      corresponding to each <ServiceRequest> -> <SituationExchangeRequest>.
    -->
    <SituationExchangeDelivery version="2.0">
      <!--
        * Count: exactly 1.
        Timestamp of generation of the response by the server.
      -->
      <ResponseTimestamp>2020-02-18T12:30:30+0000</ResponseTimestamp>

      <!--
        * Count: exactly 1.
        Contains a sequence of <PtSituationElement> elements.
      -->
      <Situations>
        <!--
          * Count: 0, 1 or many.
          The set of <PtSituationElement> elements corresponding to the
          <SituationExchangeRequest> as described by SIRI SX: Part 5.
        -->
        <PtSituationElement>
          <CreationTime>2020-02-18T11:46:25Z</CreationTime>
          <ParticipantRef>ItoWorld</ParticipantRef>
          <SituationNumber>RGlzcnVwdGlvbk5vZGU6MjQ2OA==</SituationNumber>
          <Version>1</Version>
          <Source>
            <SourceType>feed</SourceType>
            <TimeOfCommunication>2020-02-18T11:48:50Z</TimeOfCommunication>
          </Source>
          <Progress>open</Progress>
          <ValidityPeriod>
            <StartTime>2020-02-18T20:00:00Z</StartTime>
            <EndTime>2020-02-25T10:00:00Z</EndTime>
          </ValidityPeriod>
          <Repetitions>
            <DayType>wednesday</DayType>
            <DayType>thursday</DayType>
          </Repetitions>
          <PublicationWindow>
            <StartTime>2020-02-18T11:48:00Z</StartTime>
            <EndTime>2020-02-25T23:59:59Z</EndTime>
          </PublicationWindow>
          <EnvironmentReason>flooding</EnvironmentReason>
          <Planned>true</Planned>
          <Summary overridden="true" xml:lang="EN">winds in leeds city
centre</Summary>
```

```
          <Description overridden="true" xml:lang="EN">The storm Dennis has caused
some disruption and flooding</Description>
          <InfoLinks>
            <InfoLink>
              <Uri/>
            </InfoLink>
          </InfoLinks>
          <Consequences>
            <Consequence>
              <Condition>unknown</Condition>
              <Severity>slight</Severity>
              <Affects>
                <Operators>
                  <AffectedOperator>
                    <OperatorRef>FLDS</OperatorRef>
                    <OperatorName xml:lang="EN">First Leeds</OperatorName>
                  </AffectedOperator>
                </Operators>
                <Networks>
                  <AffectedNetwork>
                    <VehicleMode>bus</VehicleMode>
                    <AllLines/>
                  </AffectedNetwork>
                </Networks>
              </Affects>
              <Advice>
                <Details xml:lang="EN">flooding in leeds</Details>
              </Advice>
              <Blocking>
                <JourneyPlanner>false</JourneyPlanner>
              </Blocking>
              <Delays>
                <Delay>PT55M</Delay>
              </Delays>
            </Consequence>
          </Consequences>
        </PtSituationElement>

        <!-- ... more <PtSituationElement> elements if such are available ... -->
      </Situations>
    </SituationExchangeDelivery>
  </ServiceDelivery>
</Siri>
```

3 Streaming API reference

3.1 Subscriptions

3.1.1 <SubscriptionRequest> request

```
<Siri>
  <!--
    * Count: exactly 1.
    Request for subscription delivery of data over webhooks.
  -->
  <SubscriptionRequest>
    <!--
      * Count: exactly 1.
      Timestamp of generating the request specified by the client.
    -->
    <RequestTimestamp>2020-02-18T15:37:06+0000</RequestTimestamp>

    <!--
      * Count: exactly 1.
      The app_id of the Open Data Hub 3scale application should be put here.
      In case it is not valid or different to the app_id used in the HTTP Authorize
      header, an error would be returned.
    -->
    <RequestorRef>d0ab77fe</RequestorRef>

    <!--
      * Count: exactly 1.
      Address to which <ServiceDelivery> data delivery notifications, resulting
      from this subscription, are to be sent to.
    -->
    <ConsumerAddress>http://example.com/siri/sx/consumer</ConsumerAddress>

    <!--
      * Count: 0, 1 or many.
      Request for a subscription to the Situation Exchange (SX) service.
    -->
    <SituationExchangeSubscriptionRequest>
      <!--
        * Count: exactly 1.
        A unique identifier of the SX request generated by the client.
      -->
      <SubscriptionIdentifier>qwerty123456</SubscriptionIdentifier>
```

```
      <!--
        * Count: exactly 1.
        The app_id of the Open Data Hub 3scale application should be put here.
        In case it is not valid or different to the app_id used in the HTTP Authorize
        header, an error would be returned.
      -->
      <SubscriberRef>d0ab77fe</SubscriberRef>

      <!--
        * Count: exactly 1.
        The desired lease of the subscription as an absolute end timestamp.
        The subscription will only be granted if the lease can be met, otherwise
        an error will be returned. The maximum lease time is one week into the
        future. It is recommended that clients terminate their subscriptions
        nightly and resubscribe, at say 3am, in order to pick up the latest server
        settings and avoid connections going stale.
      -->
      <InitialTerminationTime>2020-03-01T12:30:00+0000</InitialTerminationTime>

      <!--
        * Count: 0 or 1.
        If "true", the producer would normally provide the subset of
        <PtSituationElement> elements which have changed since the last time there
        was a change. On the first <ServiceDelivery> notification after a subscription
        is established, ALL the applicable <PtSituationElement> elements for the
        subscription request will be supplied, respecting all request settings and
        filters.
        Only "true", i.e. always use incremental updates, is supported.
        Defaults to "true". "false" is not supported.
      -->
      <IncrementalUpdates>true</IncrementalUpdates>

      <!--
        * Count: 0 or 1 or many.

        Supported semantics are the same as the <SituationExchangeRequest>
        element defined in the 6.1 Polling API section.
      -->
      <SituationExchangeRequest>
        <!-- ... as defined above ... -->
      </SituationExchangeRequest>
    </SituationExchangeSubscriptionRequest>
  </SubscriptionRequest>
</Siri>
```

3.1.2 <SubscriptionResponse> response

```
<Siri>
  <!--
    * Count: exactly 1.
    Response from inform whether the requested subscriptions in the corresponding
    <SubscriptionRequest> have been created.
  -->
  <SubscriptionResponse>
    <!--
      * Count: exactly 1.
      Timestamp of generation of the response by the server.
    -->
    <ResponseTimestamp>2020-02-18T12:30:30+0000</ResponseTimestamp>

    <!--
      * Count: exactly 1.
      Producer ref corresponding to the vendor.
    -->
    <ResponderRef>TransportAPI</ResponderRef>

    <!--
      * Count: 1 or many.
      Contains information about the processing each individual service
      subscription request – either success info or error conditions. One
      <ResponseStatus> for each <SituationExchangeSubscriptionRequest>.
    -->
    <ResponseStatus>
      <!--
        * Count: exactly 1.
        Timestamp of generation of the response by the server.
      -->
      <ResponseTimestamp>2020-02-18T12:30:30+0000</ResponseTimestamp>

      <!--
        * Count: exactly 1.
        The value of the corresponding <SubscriberRef> element in the request used
        to create this subscription.
      -->
      <SubscriberRef>d0ab77fe</SubscriberRef>

      <!--
        * Count: exactly 1.
        The value of the corresponding <SubscriptionIdentifier> element in the
        request used to create this subscription.
      -->
```

```
      <SubscriptionRef>qwerty123456</SubscriptionRef>

      <!--
        * Count: Exactly 1.
        Whether the request could be processed successfully or not.
        true if successful. If false, then error conditions would be present below.
      -->
      <Status>true</Status>
    </ResponseStatus>
  </SubscriptionResponse>
</Siri>
```

3.2 Data delivery notifications

3.2.1 <ServiceDelivery> data delivery notification request

Follows the semantics of `6.2 <ServiceDelivery> response` with the following additions within the `<SituationsExchangeDelivery>` element:

```
<Siri>
  <ServiceDelivery>
    <SituationExchangeDelivery>
      <!--
        Count: exactly 1.

        The value of the corresponding <SubscriberRef> element in the request used
        to create this subscription.
      -->
      <SubscriberRef>d0ab77fe</SubscriberRef>

      <!--
        Count: exactly 1.

        The value of the corresponding <SubscriptionIdentifier> element in the
        request used to create this subscription.
      -->
      <SubscriptionRef>qwerty123456</SubscriptionRef>

      <!-- remaining elements as described in “6.2 <ServiceDelivery> response”-->
    </SituationExchangeDelivery>
  </ServiceDelivery>
</Siri>
```

3.2.2 Response

No specific response is required by the consumer: the server ignores the responses, including their status codes, headers and bodies.

3.3 Heartbeat notifications

3.3.1 <HeartbeatNotification> heartbeat notification request

```
<Siri>
  <!--
    * Count: exactly 1.
    Response from producer to consumer to inform if the service is working.
    Sent at regular intervals.
  -->
  <HeartbeatNotification>
    <!--
      * Count: exactly 1.
      Timestamp of generation of the request by the server.
    -->
    <RequestTimestamp>2020-03-20T14:06:33+0000</RequestTimestamp>

    <!--
      * Count: exactly 1.
      Producer ref corresponding to the vendor.
    -->
    <ProducerRef>TransportAPI</ProducerRef>

    <!--
      * Count: exactly 1.
      Unique server-generated identifier of the response for audit and
      investigation purposes.
    -->
    <MessageIdentifier>abcdef123456</MessageIdentifier>

    <!--
      * Count: exactly 1.
      Whether the service is available. true if available, false if not available.
    -->
    <Status>true</Status>

    <!--
      * Count: exactly 1.
      Timestamp of creation of the subscription.
```

```
    -->
    <ServiceStartedTime>2020-01-21T09:49:59+0000</ServiceStartedTime>

    <!--
      * Count: exactly 1.
      Timestamp of expiration of the subscription.
    -->
    <ValidUntil>2020-03-20T14:06:53+0000</ValidUntil>
  </HeartbeatNotification>
</Siri>
```

3.3.2 Response

No specific response is required by the consumer: the server ignores the responses, including their status codes, headers and bodies.

3.4 Unsubscriptions

3.4.1 <TerminateSubscription> request

```
<Siri>
  <TerminateSubscriptionRequest>
    <!--
      * Count: exactly 1.
      Timestamp of generating the request specified by the client.
    -->
    <RequestTimestamp>2020-02-18T15:37:06+0000</RequestTimestamp>

    <!--
      * Count: exactly 1.
      The app_id of the Open Data Hub 3scale application should be put here.
      In case it is not valid or different to the app_id used in the HTTP Authorize
      header, an error would be returned.
    -->
    <RequestorRef>d0ab77fe</RequestorRef>

    <!--
      * Count: 0 or 1.
      Terminate all subscriptions for the requestor specified via <RequestorRef>.
      Either use this element or a sequence of <SubscriptionRef> elements.
    -->
    <All/>

    <!--
```

```
      * Count: 0, 1 or many.
      Identifies a specific subscription to be terminated as specified by a
      <SubscriptionRef> element in a previous subscription request.
      Either use a sequence of these elements or <All/>.
    -->
    <SubscriptionRef>qwerty123456</SubscriptionRef>
  </TerminateSubscriptionRequest>
</Siri>
```

3.4.2 <TerminateSubscriptionResponse> response

```
<?xml version="1.0"?>
<Siri>
  <TerminateSubscriptionResponse>
    <!--
      Count: exactly 1.

      Timestamp of generation of the response by the server.
    -->
    <ResponseTimestamp>2020-02-18T12:30:30+0000</ResponseTimestamp>

    <!--
      Count: exactly 1.

      Producer ref corresponding to the vendor.
    -->
    <ResponderRef>TransportAPI</ResponderRef>

    <!--
      Count: 0, 1 or many.

      Status of each response to each subscription termination request specified
      by either <All> or a sequence of <SubscriberRef> elements in the
      corresponding <TerminateSubscriptionRequest>.
    -->
    <TerminationResponseStatus>
     <!--
        Count: exactly 1.

        Timestamp of generation of the response by the server.
      -->
      <ResponseTimestamp>2020-02-18T12:30:30+0000</ResponseTimestamp>

      <!--
        Count: exactly 1.
```

```
      The value of the corresponding <SubscriberRef> element in the request used
      to create this subscription.
    -->
    <SubscriberRef>d0ab77fe</SubscriberRef>

    <!--
      Count: exactly 1.

      The value of the corresponding <SubscriptionRef> element in the
      request used to request cancelling this subscription.
    -->
    <SubscriptionRef>qwerty123456</SubscriptionRef>

    <!--
      Count: exactly 1.
      Whether the subscription could be cancelled. true if successful.
      false if unsuccessful, in which case error conditions follow below.
    -->
    <Status>true</Status>
  </TerminationResponseStatus>
 </TerminateSubscriptionResponse>
</Siri>
```