511 Data Exchange including an Open511 Protocol

Traffic Event
Table of Contents

1  Overview.................................................................................................................................................................... 2
2  Traffic API................................................................................................................................................................... 2
   2.1  API: Event.......................................................................................................................................................... 2
3  Appendix A: API Response Messages- XML ....................................................................................................14
   3.1  Traffic XML.....................................................................................................................................................14
4  Appendix B: API Response Messages- JSON ...................................................................................................17
   4.1  Traffic JSON ...................................................................................................................................................17
List of Tables

A.1.1 Example Traffic Event Structure Response (XML).......................................................................................... 14
B.1.1 Example Traffic Event Structure Response (JSON).......................................................................................... 17
## Document History

<table>
<thead>
<tr>
<th>Description</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Draft - addressed reorganization comments</td>
<td>0.9</td>
<td>08/28/13</td>
</tr>
<tr>
<td>First published version with transit, traffic, tolling, and parking APIs</td>
<td>1.0</td>
<td>09/13/13</td>
</tr>
<tr>
<td>Update Traffic APIs’ structure information, parameters and filters, and their examples to sync with specification provided on Open511.org.</td>
<td>1.0</td>
<td>5/2/2014</td>
</tr>
<tr>
<td>Add GTFS-realtime Trip Updates and Vehicle Positions, and their examples.</td>
<td>1.0</td>
<td>5/7/2014</td>
</tr>
<tr>
<td>Minor updates and corrections</td>
<td>1.0</td>
<td>5/28/2014</td>
</tr>
<tr>
<td>Add sample request endpoint and parameters and filters tables for Section 3.14 and 3.15. Update references for resource endpoints with their exact URL.</td>
<td>1.0</td>
<td>6/12/2014</td>
</tr>
<tr>
<td>Minor updates to Section 3.14 and 3.15</td>
<td>1.0</td>
<td>7/17/2014</td>
</tr>
<tr>
<td>Split API specification document into sub docs for each API domain</td>
<td>1.0</td>
<td>8/26/2014</td>
</tr>
<tr>
<td>Minor updates to remove Transit references</td>
<td>1.0</td>
<td>9/24/2014</td>
</tr>
<tr>
<td>Removed APIs that are not planned to be developed</td>
<td>1.0</td>
<td>08/01/2017</td>
</tr>
<tr>
<td>Updated to add road closure polyline</td>
<td>1.1</td>
<td>08/19/2018</td>
</tr>
<tr>
<td>Updated to include splitting longer geometries into multiple 100 point Linestring.</td>
<td>1.2</td>
<td>11/01/2018</td>
</tr>
<tr>
<td>Updated to reflect that Linestring will always be 100 points in length and won’t be controlled by an API parameter.</td>
<td>1.3</td>
<td>02/25/2018</td>
</tr>
</tbody>
</table>
1 Overview

This document focuses on data exchange APIs for the traffic event data. For a complete overview of 511 Data Exchange, please refer to *Open 511 Data Exchange Specifications – Overview* document. The overview document covers:

- General information about 511 Data Exchange
- Different protocols and data feeds available through Open 511 APIs
- Standard Discovery API specifications.
- Encodings and Protocols along with reference to standard documentation.
- Technical Guidelines

It is highly recommended that all users of Open 511 Data Exchange review the information in the Overview document.

2 Traffic API

The core traffic data resources consist of traffic events. Open511 provides message structure and API endpoint for accessing these resources. Open511 will also provide metadata/lookup information that can be used by consumers to filter and limit information during requests.

2.1 API: Event

The event resource provides information about various types of events within a jurisdiction. These can be active incident, scheduled construction/roadway work or public events which may or may not have an impact on traffic conditions. Consumers can request list of all the active events or they can use additional filters such as city, roads to restrict the results as per their needs and use case.

The event structure is the main element of the events collection. Although an event should be considered as independent, it is possible that one major event (mainly construction) could be split across several events. Below is a message structure of Event.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self</td>
<td>Link</td>
<td><em>Mandatory</em></td>
<td>Self link to the current resource.</td>
</tr>
<tr>
<td>Jurisdiction</td>
<td>Link</td>
<td><em>Mandatory</em></td>
<td>Link to the jurisdiction publishing the event.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td><em>Mandatory</em></td>
<td>A globally unique ID for the event, following the format jurisdiction-id/event-id. For example, 511.org/8c3f2. The first segment of the event ID is the jurisdiction ID. The second segment is a</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>Free Text</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Enum</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>event_type</td>
<td>Enum</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>Severity</td>
<td>Enum</td>
<td>Mandatory</td>
<td></td>
</tr>
</tbody>
</table>

**string ID that must be unique within its jurisdiction. It can contain the characters a-zA-Z0-9_.-. The two segments are separated with a /.

The event's self link is of course itself a globally unique ID, and it is a suggested practice to use the event ID in the event URL, but a separate ID field is included for user-friendliness.

**Status**

Status of the event. The status allows a client to determine if the current event should be considered as currently effective. Value list:

- ACTIVE : The event should be considered as effective
- ARCHIVED : The event should not be considered as effective

**Headline**

Title of the event. Should be shorter than 500 characters.

**event_type**

Basic type of event. Value list

- CONSTRUCTION : planned road work
- SPECIAL EVENT : special events (fair, sport event, etc.)
- INCIDENT : accidents and other unexpected events
- WEATHER_CONDITION : Weather condition affecting the road
- ROAD_CONDITION : Status of the road that might affect travelers.

**Severity**

Severity of the event. Value list:

- MINOR: the event has very limited impact on traffic.
- MODERATE: the event will have a visible impact on traffic but should not create significant delay; if there is a delay, it should be small and local.
- MAJOR: the event will have a significant impact on traffic, probably on a large scale.
- UNKNOWN: the impact is unknown, for example in the case of an accident that has been recorded without any precise description.
<table>
<thead>
<tr>
<th><strong>Geography</strong></th>
<th>Geospatial</th>
<th>Mandatory</th>
<th>Geographical information about the event. Can be Point, MultiPoint, LineString, MultiLineString, or Polygon.</th>
</tr>
</thead>
<tbody>
<tr>
<td>closure_geometry</td>
<td>Geospatial</td>
<td>Optional</td>
<td>Multistring geometry for closure associated with an event. The individual LineStringMember represents affected segment of a road. For events that expand over non-contiguous road segments, for example a closure affecting both sides of the freeway, each side will be represented by separate LineStringMember. For continuous event, entire geometry will be split into multiple LineStringMembers of 100 points each, due to limitation on number of points that can be included in one line string. When this happens, end point of one line string will be the start point of the next line string.</td>
</tr>
<tr>
<td>Created</td>
<td>DateTime (UTC)</td>
<td>Mandatory</td>
<td>When the event was initially created.</td>
</tr>
<tr>
<td>Updated</td>
<td>DateTime (UTC)</td>
<td>Mandatory</td>
<td>When the content of the event last changed. Will be the same as created if no updates have occurred.</td>
</tr>
<tr>
<td>Schedule</td>
<td>schedule elements</td>
<td>Mandatory</td>
<td>Indicates the dates and times when the event is active.</td>
</tr>
<tr>
<td>Timezone</td>
<td>timezone</td>
<td>Optional</td>
<td>Timezone to be used for this event, e.g. America/Montreal. If not provided, the event is assumed to be in the default timezone of its jurisdiction. <em>But strongly recommended.</em></td>
</tr>
<tr>
<td>Description</td>
<td>Free text</td>
<td>Optional</td>
<td>Description of the event. Plain text description of the event, the reason for the event and any other relevant information for travelers.</td>
</tr>
<tr>
<td>event_subtypes</td>
<td>Collection of event_subtype elements</td>
<td>Optional</td>
<td>List of values to provide more detail about the type of event.</td>
</tr>
</tbody>
</table>
| — event_subtype | Enum | Optional | - ACCIDENT  
- SERIOUS_ACCIDENT  
- NUMEROUS_ACCIDENTS  
- STALLED_VEHICLE  
- SPILL  
- OBSTRUCTION |
- MAJOR_HAZARD
- DEMONSTRATION

- ROAD_CONSTRUCTION
- WORK_IN_THE_MEDIAN
- NARROW_LANES

- TRAFFIC_ALTERNATING_DIRECTIONS
- BRIDGE_OPERATIONS
- BLASTING
- WORK_ON_UNDERGROUND
- EMERGENCY_MAINTENANCE

- SPORTING_EVENT
- MAJOR_EVENT
- CONCERT
- FESTIVAL
- FIREWORKS
- PARADE
- CROWD

- SEVERE_WEATHER
  - HEAVY_SNOW
  - SNOW
  - ICE_GLAZE
  - HEAVY_FROST
  - ICE_STORM
  - DAMAGING_HAIL
  - THUNDERSTORM
  - HEAVY_DOWNPOUR
  - TORNADO
  - HURRICANE
  - STRONG_WINDS
  - DENSE_FOG
  - FREEZING_FOG
  - ICE_FOG_MIST

- VISIBILITY_REducted
  - VISIBILITY_BLOCKED
- BLOWING_SNOW
- BLOWING_DUST
- SANDSTORM
- INSECT_SWARMS

- IMPASSABLE
  - ALMOST_IMPASSABLE
  - PASSABLE_WITH_CARE
- SURFACE_WATER_HAZARD
- HYDROPLANING_DANGER
- SLIPPERY
- MUD
- LOOSE_GRAVEL
- OIL_ON_ROADWAY
- ICE
- BLACK_ICE
- WET_ICY_ROAD
- SLUSH
### Certainty

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enum</td>
<td>Degree of certainty of the event. Should only be used for unplanned events (e.g. incidents, weather conditions and pavement conditions events). Value list to be confirmed. Could be observed/likely/possible/unknown.</td>
</tr>
<tr>
<td>Optional</td>
<td>Value list</td>
</tr>
<tr>
<td></td>
<td>- OBSERVED</td>
</tr>
<tr>
<td></td>
<td>- LIKELY</td>
</tr>
<tr>
<td></td>
<td>- POSSIBLE</td>
</tr>
<tr>
<td></td>
<td>- UNKNOWN</td>
</tr>
</tbody>
</table>

### grouped_events

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection of links</td>
<td>This structure is used to group events together. In specific situations (for example complex construction projects), several events might be related together. This field should be used to point a related event.</td>
</tr>
<tr>
<td>Optional</td>
<td></td>
</tr>
</tbody>
</table>

### related

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Link pointing to another event resource related to the current event.</td>
</tr>
<tr>
<td>Optional</td>
<td></td>
</tr>
</tbody>
</table>

### Detour

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free Text</td>
<td>Description of a detour route to avoid this event.</td>
</tr>
<tr>
<td>Optional</td>
<td></td>
</tr>
</tbody>
</table>

### Roads

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection of road elements</td>
<td>List of roads affected by the current event. One event can impact several roads.</td>
</tr>
<tr>
<td>Optional</td>
<td></td>
</tr>
</tbody>
</table>

### Areas

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collections of area elements</td>
<td>Areas affected by the event.</td>
</tr>
<tr>
<td>Optional</td>
<td></td>
</tr>
</tbody>
</table>

### attachments

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection of attachment links</td>
<td>Collection of attachments providing additional information about the event (PDFs, images, etc.)</td>
</tr>
<tr>
<td>Optional</td>
<td></td>
</tr>
</tbody>
</table>

### related

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Link to an attachment. While only the URL is mandatory, more information may be provided via type, length, title, and hreflang, with semantics as in Atom.</td>
</tr>
<tr>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
</tr>
<tr>
<td>Name</td>
<td>Free Text</td>
</tr>
<tr>
<td>self / url</td>
<td>Link</td>
</tr>
<tr>
<td>from</td>
<td>Free text</td>
</tr>
<tr>
<td>To</td>
<td>Free text</td>
</tr>
</tbody>
</table>
| State     | Enum         | **Optional**       | Whether the road segment is closed or not. **Value list:**  
- CLOSED (road closed in the given direction)  
- SOME_LANES_CLOSED (but the road remains open)  
- SINGLE_LANE_ALTERNATING (a single lane alternates between both directions of traffic)  
- ALL_LANES_OPEN |
| direction | Enum         | **Conditional**    | *Mandatory if state is provided.* Direction of the road that is affected by the event.  
If a lane_status is provided, the direction becomes mandatory. In the same situation, if both directions are affected, two occurrences of the road element are needed, one for each direction and with a dedicated lane_status for each direction. **Value list:**  
- N |
### Restriction Structure

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Mandatory/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>restriction_type</td>
<td>Enum</td>
<td>Mandatory</td>
<td>Type of restriction that affects vehicles. Value list:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPEED: Limitation of the speed of vehicles. Unit is in kilometers/hour.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WIDTH: Width limitation, mainly for trucks. Unit is meters.</td>
</tr>
</tbody>
</table>

### fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lanes_open</td>
<td>Integer</td>
<td>Optional</td>
<td>Number of lanes in the given direction remaining open during this event.</td>
</tr>
<tr>
<td>lanes_closed</td>
<td>Integer</td>
<td>Optional</td>
<td>Number of lanes in the given direction closed during this event.</td>
</tr>
<tr>
<td>impacted_system</td>
<td>Collection of impacted_system elements</td>
<td>Optional</td>
<td>Allows provision of information about other systems that can be affected.</td>
</tr>
<tr>
<td>impactsystem</td>
<td>Enum</td>
<td>Optional</td>
<td>Value list of systems:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ROAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SIDEWALK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BIKELANE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PARKING</td>
</tr>
<tr>
<td>restrictions</td>
<td>Collection of restriction elements</td>
<td>Optional</td>
<td>Some events may come with some restrictions affecting vehicles using the road (speed, weight).</td>
</tr>
</tbody>
</table>
HEIGHT: Height limitation, mainly for trucks. Unit is meters
WEIGHT: Weight limitation for vehicles. Unit is metric tons.
AXLE_WEIGHT: Weight limitation per axle for truck. Unit is metric tons.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Mandatory/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Float</td>
<td>Mandatory</td>
<td>Value of the limitation. For example, a speed limitation of 60km/h will have a value of 60 with a restriction_type set to SPEED.</td>
</tr>
</tbody>
</table>

**Schedule structure**

The schedule defines timelines of an event.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Mandatory/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recurring_schedules</td>
<td>Collection of recurring_schedule elements</td>
<td>Conditional</td>
<td>One (and only one) of recurring_schedules or intervals is required. The recurring_schedule structure expresses repeating schedules, like &quot;Every day starting December 4th&quot;, or &quot;Mondays 9 to 11 from September 1 to October 30.&quot; An event can include override them.</td>
</tr>
<tr>
<td>— start_date</td>
<td>Date</td>
<td>Mandatory</td>
<td>Mandatory in each recurring_schedule. Start date of this schedule.</td>
</tr>
<tr>
<td>— end_date</td>
<td>Date</td>
<td>Optional</td>
<td>End date of the event. If a start date but no end date is provided, the schedule continues indefinitely.</td>
</tr>
<tr>
<td>— daily_start_time</td>
<td>Time</td>
<td>Optional</td>
<td>Daily start time of the event, as HH:mm, e.g. 13:00. Applies to each day in this recurring_schedule.</td>
</tr>
<tr>
<td>— daily_end_time</td>
<td>Time</td>
<td>Conditional</td>
<td>Mandatory if daily_start_time is provided, not allowed otherwise.</td>
</tr>
<tr>
<td><strong>— days</strong></td>
<td>collection of <em>day</em> elements</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>exceptions</strong></td>
<td>Collection of <em>exception</em> elements</td>
<td>Optional</td>
<td></td>
</tr>
<tr>
<td><strong>— exception</strong></td>
<td>Custom time format</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td><strong>intervals</strong></td>
<td>Collection of <em>interval</em> elements</td>
<td>Conditional</td>
<td></td>
</tr>
</tbody>
</table>

**Daily end time of the event, as HH:mm, e.g. 17:30.** Applies to each day in this *recurring_schedule*.

Contains a *day* tag for every day of the week during which this *recurring_schedule* is active. Days are indicated with an integer, with (following the ISO standard) Monday being 1 and Sunday 7. So, for an event active on Monday and Wednesday, `<days><day>1</day><day>3</day></days>`. If omitted, the schedule is active every day between its start and end dates.

*If present, there must be a* *recurring_schedules* *element (and no* intervals*)

Exceptions override recurring schedules.

An exception provides the definitive schedule for a specific date. It overrides any *recurring_schedule* information for that date.

An exception of the form YYYY-MM-DD indicates that this event is **not** in effect for the given date.

An exception of the form YYYY-MM-DD HH:mm-HH:mm indicates that, on that date, the event is in effect only between the provided start and end time. If there are multiple disjoint periods on that day, more than one start-end period can be included: YYYY-MM-DD HH:mm-HH:mm HH:mm-HH:mm.
Represent an event's schedule as a list of explicit periods. An event must use either `recurring_schedules` or `intervals`, not both.

If this contains more than one `interval`, their time periods may not overlap. This also implies that no more than one can omit an end time.

Defines a specific period of time during which the schedule is in effect; two `datetimes` (without timezone offset) joined with a `/`. For example, a period from 9 p.m. September 1 to 8 a.m. September 2 would be `2014-09-01T21:00/2014-09-02T08:00`.

The second datetime, after the slash, may be omitted. So, to indicate a period from 9 p.m. September 1 until further notice, use: `2014-09-01T21:00/`.

---

Sample request endpoint for events

<table>
<thead>
<tr>
<th>Request Type</th>
<th>GET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Endpoint Example</td>
<td>For e.g. <a href="http://api.511.org/Traffic/Events">http://api.511.org/Traffic/Events</a></td>
</tr>
</tbody>
</table>

Parameters and Filters supported with the request

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandatory/ Optional</th>
<th>Description</th>
</tr>
</thead>
</table>
| format    | Optional            | The response format (json/xml) desired. If none specified, then default response would be JSON.
|           |                     | e.g.        |
|           |                     | ?format=json |
|           |                     | (returns json response for v1, if v1 is the latest version or specified via version parameter) |
|           |                     | ?format=xml  |
| **version** | Optional | The version of Open511 desired.  
| e.g. | ?version=v1  
| (returns response for v1 in conjunction with format requested. |

| **api_key** | Mandatory | Unique key assigned to a user after they signup for Open511. |

| **status** | Optional | By default the API should only send active events. Supported values:  
| | ACTIVE Default value, returns only active events.  
| | ARCHIVED Returns only archived events  
| | ALL Returns both active and archived events. |

| **in_effect_on** | Optional | Show only events that are, according to their schedules, in effect at a specific time, or during a specific time period.  
| | Can be either a single time, or a start and end time joined by a comma. The times must be complete ISO 8601 datetimes, with or without a timezone.  
| | So to find, for example, all events in effect at some point on June 20th, you would ask for events between 00:00 and 23:59:  
| | in_effect_on=2013-06-20T00:00,2013-06-20T23:59  
| | Or, to find events in effect within the next two hours, you'd get the current UTC timestamp — let's say it's 2013-06-20T17:40Z — and then request  
| | You can also use the special value ?in_effect_on=now to show events currently taking place.  
| | If no timezone is provided, as in the first example, the server searches according to the local time as entered in the event. For example, if highways are closed for an hour at midnight on January 1st in both London and LA, ?in_effect_on=2014-01-01T00:00 would return both events (even though there's no single moment in time when the roads are closed in both cities), whereas the timezone-aware  
| | ?in_effect_on=2014-01-01T00:00Z would return only the London event.  
| | If this argument is not provided, the API should not perform any filtering based on the event schedules. |

| **severity** | Optional | Filter by the value of the severity field. OR queries are possible via a comma-separated list: ?severity=MINSOR,MODERATE. |

| **Bbox** | Optional | Filter events by geographical bounding box. Four comma-separated coordinates, xmin, ymin, xmax, ymax. |

| **geography** | Optional | Filtering near a point or linestring. The geography value should be provided as a WKT string in WGS84 latitude/longitude. Example:  
| | ...geography=POINT (-73.64 45.52).  
| | The geography must be used in conjunction with the tolerance filter. |
|                | Conditional | Provide a tolerance of radius in meters around if a POINT or LINESTRING is used as filtering parameter. For example ...
|----------------|-------------|--------------------------------------------------------------------------------|
|                | Conditional | ...
|                |             | geography=POINT+(-73.64+45.52)&tolerance=50 would retrieve all the events within a circle of 50m radius around the selected point. |
| **jurisdiction** | Optional    | The ID or URL of a jurisdiction, in order to show only events from a given jurisdiction. Use a comma-separated list to make OR queries. |
| **event_type**  | Optional    | Filter events using the event_type value list. Use a comma-separated list to make OR queries. |
| **event_subtype** | Optional    | Filter events using the event_subtype value list. Use a comma-separated list to make OR queries. |
| **created**     | Optional    | Filter events based on the creation date and time. |
| **updated**     | Optional    | Filter events based on the last update timestamp. Accepts the same < <= > >= operators as created. |
| **road_name**   | Optional    | Shows only events containing a road element with the provided name. Case-sensitive exact match. Use a comma-separated list to make OR queries. |
| **Road**        | Optional    | Shows only events containing a road element linking to a road element with the provided id. Use a comma-separated list to make OR queries. |
| **Area**        | Optional    | Shows only events containing an area element with the provided id. Use a comma-separated list to make OR queries. |
The traffic event structure response for XML is shown in Appendix A Section A.I.1 and for JSON is shown in Appendix B Section B.I.1.

Possible Errors

Listed below are HTTP status code and message returned for certain common errors:

- 500 - Internal Server Error (System has issues processing your request)
- 401 – Unauthorized (Invalid API key)
- 404 – Not found (If an individual event resource cannot be located).

3 Appendix A: API Response Messages- XML

3.1 Traffic XML

A.I.1 Example Traffic Event Structure Response (XML)

```xml
<open511
    xmlns:gml="http://www.opengis.net/gml"
    xml:lang="en"
    xml:base="http://api.511.org"
    version="v1"
>
<events>
  <event>
    <status>ACTIVE</status>
    <link rel="self" href="/traffic/events/511.org/149"/>
    <link rel="jurisdiction" href="http://api.511.org/jurisdictions/511.org/"/>
    <id>511.org/149</id>
    <headline>CHP : Roadwork on CA-160 NorthBound between Main St (Antioch) and Antioch Bridge - Toll Plaza (Oakley) Acceleration lane closed Expect delays</headline>
    <event_type>INCIDENT</event_type>
    <severity>UNKNOWN</severity>
    <created>2014-05-01T19:28:31Z</created>
    <geography>

```
<gml:Point srsName="EPSG:4326">
  <gml:coordinates>-121.75382399999999,38.004908</gml:coordinates>
</gml:Point>

<geography>
<roads>
<road>
  <name>CA-160</name>
  <from>Main St</from>
  <to>Antioch Bridge - Toll Plaza</to>
  <direction>NorthBound</direction>
  <state>Open</state>
</road>
</roads>
</geography>

<schedules>
<schedule>
  <start_date>2014-05-01</start_date>
</schedule>
</schedules>

<event>
  <status>ACTIVE</status>
  <link rel="self" href="/traffic/events/511.org/209"/>
  <link rel="jurisdiction" href="http://api.511.org/jurisdictions/511.org/"/>
  <id>511.org/209</id>
  <headline>CHP : Obstruction on US-101 N NorthBound before Coyote Creek Golf Dr (San Jose) left lane blocked Expect delays</headline>
  <event_type>INCIDENT</event_type>
  <severity>UNKNOWN</severity>
  <created>2014-05-02T01:13:55Z</created>
  <updated>2014-05-02T02:43:16Z</updated>
  <geography>
    <gml:Point srsName="EPSG:4326">
      <gml:coordinates>-121.69346399999999,37.19068</gml:coordinates>
    </gml:Point>
  </geography>
  <roads>
    <road>
      <name>US-101 N</name>
      <from>Coyote Creek Golf Dr</from>
      <to/>
      <direction>NorthBound</direction>
      <state>Open</state>
    </road>
  </roads>
  <schedules>
    <schedule>
      <start_date>2014-05-01</start_date>
    </schedule>
  </schedules>
</event>

<closure_geometry xmlns:gml="http://www.opengis.net/gml" xmlns="http://511.org/open511-extensions">
<gml:MultiLineString srsName="urn:ogc:def:crs:EPSG::4326">
  <gml:LineStringMember>
    <gml:LineString>
      <gml:posList>-121.905403778982,37.4664031808211,-121.905516999724,37.4665420000284,-121.905530999967,37.466599999776,-121.9055500000234,37.4665840000328,-121.9056659999391,37.4667269999391,-121.906012000191,37.4671670002383,-121.906404999653,37.4676470002688,-121.906715000173,37.4680369998512,-121.905403778982,37.4664031808211</gml:posList>
    </gml:LineString>
  </gml:LineStringMember>
</gml:MultiLineString>
</closure_geometry>
January 1, 2019 Page 16 of 22
4 Appendix B: API Response Messages - JSON

4.1 Traffic JSON

B.1.1 Example Traffic Event Structure Response (JSON)

```json
{
  "events": [
    {
      "url": "/traffic/events/511.org/149",
      "jurisdiction_url": "http://api.511.org/jurisdictions/511.org/",
      "id": "511.org/149",
      "status": "ACTIVE",
      "headline": "CHP : Roadwork on CA-160 NorthBound between Main St (Antioch) and Antioch Bridge - Toll Plaza (Oakley) Acceleration lane closed Expect delays",
      "event_type": "INCIDENT",
      "severity": "UNKNOWN",
      "created": "2014-05-01T19:28:31Z",
      "geography": {
        "type": "Point",
        "coordinates": [
          -121.753824,
          38.004908
        ]
      },
      "roads": [
        {
          "name": "CA-160",
          "from": "Main St",
          "to": "Antioch Bridge - Toll Plaza",
          "direction": "NorthBound",
          "state": "Open"
        }
      ],
      "schedules": [
        {
          "start_date": "2014-05-01"
        }
      ],
      "+closure_geography": {
        "type": "MultiLineString",
        "coordinates": [
          [-121.965344339091, 38.0227069562185],
          [-121.965265000299, 38.0226909996807],
          [-121.964901000251, 38.0226219996731],
          [-121.963347999975, 38.0223280002051],
          [-121.963122999984, 38.0222860001848]
        ]
      }
    }
  ]
}
```
511 Data Exchange Specification – Traffic Event API

`,
[  -121.962568,
   38.0216900001024
],
[  -121.962037000446,
   38.0220539997063
],
[  -121.961545000351,
   38.0219580000743
],
[  -121.960780000445,
   38.0218160002701
],
[  -121.960453000207,
   38.0217570002606
],
[  -121.960274000107,
   38.0217250000316
]
],
]
``
},
{
"url": "/traffic/events/511.org/209",
"jurisdiction_url": "http://api.511.org/jurisdictions/511.org/",
"id": "511.org/209",
"status": "ACTIVE",
"headline": "CHP : Obstruction on US-101 N NorthBound before Coyote Creek Golf Dr (San Jose) left lane blocked Expect delays",
"event_type": "INCIDENT",
"severity": "UNKNOWN",
"created": "2014-05-02T01:13:55Z",
"updated": "2014-05-02T02:43:16Z",
"geography": {
"type": "Point",
"coordinates": [
-121.693464,
37.19068
]
},
"roads": [{
"name": "US-101 N",
"from": "Coyote Creek Golf Dr",
"to": "",
"direction": "NorthBound",
"state": "Open"
}],
"schedules": [{
"start_date": "2014-05-01"
}]`
"pagination": {
    "next_url": "/traffic/events/?api_key={api_key}&limit=2&offset=2",
    "offset": 0
},
"meta": {
    "url": "traffic/events/?api_key={api_key}&limit=2&offset=0",
    "up_url": "/",
    "version": "v1"
}