Travel Time Data Service

Mobile API

Document Control

Revision History

Version	Author	Issue Date	Changes
1	TfNSW	15/04/2016	Initial version

Travel Time Data Service (TTDS) Mobile API

Introduction

Roads and Maritime Services (RMS) and Transport for New South Wales have developed TTDS to allow developers to build apps from real-time traffic data.

Data is collected in real-time from commercial vehicles to give the most up to date data.

Note: This API is restricted to calls from mobile devices and due to CORS restrictions cannot be guaranteed to work with web applications executed through web browsers.

Audience

The intended audience is mobile application developers.

Keywords

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and “OPTIONAL” in this document are to be interpreted as described in RFC 2119 [1].

Mobile Application API Overview

The Mobile Application API is based on HTTP 1.1, and supports four HTTP requests:

API	Description
Route	Input: Route path and departure/arrival time. Output: Route identifier, travel time, current events on that path.
Progress	Input: Route identifier, timestamp, speed and heading. Output: Travel time, current events on the route path.
Events	Output: All events on the road network.

Note: The TTDS will not provide route finding capabilities. Mobile application developers will need to use a third party service for this purpose.

Definitions

Term	Definition
API	Application Programming Interface
Bearing	Direction of vehicle movement in degrees
Bounding Box	A bounding box is an area defined by two longitudes and two latitudes
Client Application	An application that interacts with the TTDS HTTP API
Cordova	Apache Cordova is the software underlying the PhoneGap sample application
CORS	Cross-Origin Resource Sharing
Current Data	Data which are less than 15 minutes old
Epoch	Time since January 1st 1970 00:00:00 UTC/GMT
Event	A traffic congestion event
GMT	Greenwich Mean Time
GPS	Global Positioning System
GZip	Compression utility
Historical Data	Data which are more than 15 mins old
HTTP	Hypertext Transfer Protocol
JDBC	Java Database Connectivity
JQuery	A cross-platform JavaScript library designed to simplify the client-side scripting of HTML
JQuery Mobile	A touch-optimised mobile framework developed by the JQuery project team
JSON	JavaScript Object Notation
lat	Latitude
Link	A grouping of Road Elements

lng	Longitude
Mobile Apps	See Client Application
PhoneGap	An open source cross-platform platform for building native mobile applications using HTML, CSS and JavaScript. PhoneGap is a distribution of Cordova.
Polyline	Contiguous line made up of one or more line segments.
Progress	The activity of travelling down a Route.
Rapid Acceleration Event	A point on the road network where slow moving traffic is immediately followed by fast moving traffic.
Rapid Deceleration Event	A section of road where average speeds have fallen by 15% per minute for two minutes in a row.
RE	Road Element. These are on average 10 metres in length.
REID	Road Element identifier
RMS	Roads and Maritime Services
Route	Path from a starting point to destination
Segment	A section of road that is identified separately in an asset register.
SSL	Secure Sockets Layer
TfNSW or TNSW	Transport for New South Wales
TLS	Transport Layer Security
TMC	Transport Management Centre who monitor and manage the road network
TTDS	Travel Time Data Service
TTDS Enabled Handset	Handset device using a Client Application
URI	Uniform Resource Identifier

WGS84	The World Geodetic System (WGS) is a standard for use in cartography, geodesy and navigation. WGS 84 is the reference coordinate system used by Global Positioning Systems.

System Behaviour

Client Behaviour

User-Agent header

All HTTP requests MUST include a User-Agent header that contains details of the application making the request, as per the following BNF grammar:

```
<app-version> ::= <major-version> “.” <minor-version>
<user-agent-header> ::= "User-Agent: " <application-name> “/” <app-version>
```

Accept-Encoding header

If the mobile application can accept a GZipped response then it must supply an Accept-Encoding header with the value “gzip” in the request. (e.g. The Events request.) Provision of an Accept-Encoding header with the value “gzip” in a request MUST NOT be interpreted to mean that the response will be GZipped.

Server Behaviour

Authentication

All HTTP requests MUST be authenticated.

Transport Layer Security

All HTTP requests must be received over a TLS connection. The TTDS MUST use SSL sessions on order to reduce the overhead of establishing TLS connections for multiple HTTPS requests coming from one mobile device.

Content-Type header

All POST requests MUST include a Content-Type header with an appropriate value. See the details of each individual HTTP request for the appropriate header value. All HTTP responses that have body content MUST include a Content-Type header with an appropriate value. See the details of each individual HTTP request for the appropriate header value.

HTTP Status Codes

Below are the descriptions of some HTTP status codes that are common across the API.

400 Bad Request

When the TTDS is unable to parse, or understand, the message body in a request, or the request is missing mandatory headers, the service:

- MUST return a 400 status code in the response.

401 Unauthorized

When the TTDS receives a request containing an unauthorised username, or a password that does not match an authorised username, the service:

- MUST return a 401 status code in the response.
- MUST NOT provide any indication of whether the authentication issue was due to the supplied username or password.

405 Method Not Allowed

When the HTTP method specified in a request is not supported for the specified resource, the service:

- MUST return a 405 status code in the response, and
- SHOULD include an Allow header in the response containing a list of valid methods for the requested resource.

500 Internal Service Error

When TTDS receives a request and is unable to respond. The service is unable to respond due to a failure or timeout.

- MUST return a 500 status code in the response

503 Service Unavailable

When any service is unable to handle a request due to system issues (overloaded, unable to respond, etc), the service:

- MUST return a 503 status code in the response, and
- SHOULD include an Retry-After header in the response specifying when the client application should retry the request. (This allows the service to control load generated by retries, assuming the client application honours the Retry-After value.)

504 Gateway Timeout

TTDS clients may receive error 504 due to conditions in the network. The error is not returned by the TTDS server but rather by the network. It reflects a transient condition in the network which will be corrected by the Network Operations Center or TTDS Support Services.

Additional Features

Two additional features are present in the TTDS API. These include some debugging information returned as part of the XX-Error-Msg response, and the requirement for Cookies. Both are explained here.

Invalid API Calls

To ensure security of the server, calls to invalid APIs e.g. “/notanapi” will not have access to the application layer, as they will be blocked by security controls before reaching it. Therefore no error response will be sent to the caller from the TTDS API.

XX-Error-Msg

An additional response header can be returned in the HTTP header from */route*, */progress* and */events* called *XX-Error-Msg*. When an HTTP error status is returned, this response header contains a message indicating the cause of the problem that may be helpful either to the mobile application developer or to the team supporting TTDS. This can safely be ignored by applications but the application developer can choose to log this to assist in resolving issues.

Cookies

The TTDS API servers send a cookie to clients to associate the client with a particular server for the duration of their session.

Mobile Application API

Where HTTP headers are mentioned below, those specific headers MUST be included in the request or response. However, that does not imply that these are the only headers to be included in a request or response. **Note:** The request and response messages shown in this section are examples only.

Travel Time

Travel time is calculated using actual speeds on the road. If no real time speed data is available then historical speed data is used to calculate the travel time.

Events

The Events API captures two types of events in real-time - Rapid Acceleration ("acceleration" in the response) and Rapid Deceleration ("deceleration" in the response). Speeds are captured from the real-time GPS data stream that is provided to the TTDS API from vehicles and mobile phone applications. The TTDS API monitors the network for any rapid changes in speed, and when a set of criteria are met, the event is captured as a rapid acceleration event or rapid deceleration event.

Events may be triggered by a variety of conditions such as accidents, roadworks, special events or congestion to name a few.

Road Network

The road network consists of all roads in New South Wales and Australian Capital Territory.

1. Highway name or road name is the sign posted name of the road.
2. Speed Limit is the speed limit in Kilometres per hour.
3. Bearing is a value in the range 0<= bearing < 360 degrees. Bearing 0 is North, 90 is East, 180 is South, and 270 is West.

API Details

Route API

This request enables mobile applications to query the system for the current travel time for a single route path, where the details describing the route path are supplied by the third party application. Routes are cached until later of (a) 60 minutes after the expected journey end time or (b) 60 minutes after the last */progress* API call for the route independent of the travel time. Route expiration is not intended as a precise timer, routes are cleaned up lazily, but not earlier than the specified 60 minute period

Route Response

POST

/route

Response Codes

- 200 **OK**

- 400 **Bad Request**: If the request contains bad content or badly formed content (e.g. poorly formed JSON, missing fields, no request message, invalid polyline, polyline outside of NSW/ACT geography, polyline not on road) or in the case of a route matching failure. Where a route matching failure has occurred, the caller should also see in the response an XX-Error-Message = "route mapping failed".
- 401 **Unauthorized**
- 500 **Internal Service Error**
- 503 **Service Unavailable**
- 504 **Gateway Timeout**

- **Request**Hide
- Headers

```
Content-Type: application/vnd.ttds-route+json
Body
{
  "encoded-paths": [
    "<encoded-polyline-string>",
    "<encoded-polyline-string>",
    "<encoded-polyline-string>"
  ],
  "departure-time": 1399437830,
  "provide-events": true
}
OR
{
  "encoded-paths": [
    "<encoded-polyline-string>"
  ],
 "arrival-time": 1399437830,
 "provide-events": true
}
Schema
{
  "encoded-paths": {
    "type": "Array of String",
    "required": true,
    "description": "An array of one or more polylines representing the
entire route path of the end user's journey. The encoded polylines must be
in Google Encoded Polyline Algorithm Format [6]. Subsequent polyline strings
must represent adjacent connected portions of the route. i.e. Polyline 1: A
to B, Polyline 2: B to C."
  },
  "departure-time": {
    "type": "Number",
    "required": false,
    "description": "The time the end user is proposing to start their
journey on the specified route path, in seconds since epoch in GMT. Either a
*departure-time* or *arrival-time* value MUST be provided. If both
*departure-time* and *arrival-time* are provided then *arrival-time* will be
ignored. If neither is provided the default is that *departure-time* is set
to the current time as received by the TTDS service."
  },
  "arrival-time": {
    "type": "Number",
    "required": false,
    "description": "The time the end user wishes to end their journey on the
specified route path, in seconds since epoch. Either a *departure-time* or
*arrival-time* value MUST be provided. If both *departure-time* and
*arrival-time* are provided then *arrival-time* will be ignored. If neither
is provided the default is that *departure-time* is set to the current time
as received by the TTDS service."
  },
  "provide-events": {
    "type": "Boolean",
    "required": false,
    "description": "A flag indicating whether the TTDS should include events
on the specified route path in the response message. The events are the
current events along the specified route. Default value: false."
  }
}
•    Response  200Hide
•    Headers
Cache-Control: private, max-age=60 (set to the internal travel time update
period)
Content-Type: application/vnd.ttds-traveltime+json
Set-Cookie: <See System Behaviour Section, Additional Features Sub-Section,
Cookies Paragraph, ignore in progress response>
X-Server-Hash: 1234
Body
{
  "route-id": "ab5a61f0-d99f-4769-8c8b-5d6564ef807d",
  "system-time": 1399439319,
  "data-time": 1399439319,
  "travel-time": {
    "min-seconds": 336,
    "max-seconds": 400
   }
}
OR
{
  "route-id": "ab5a61f0-d99f-4769-8c8b-5d6564ef807d",
  "system-time": 1399439319,
  "data-time": 1399439319,
  "travel-time": {
    "min-seconds": 336,
    "max-seconds": 400
  },
  "events": [
    {
      "event-id": 123,
      "type": "deceleration",
      "detection-time": 1399123400,
      "expected-end-time": 1399123400,
      "head": {
        "lng": 151.0784132,
        "lat": -33.7629486,
        "bearing": 43,
        "road-name": "M2"
      },
      "tail": {
        "lng": 151.0784132,
        "lat": -33.7629486
      },
      "congestion-backlog": {
        "length": 1000,
        "min-travel-time": 100,
        "max-travel-time": 200
      }
    },
    {
      "event-id": 124,
      "type": "acceleration",
      "detection-time": 1399345600,
      "expected-end-time": 1399345600,
      "head": {
        "lng":151.0485441,
        "lat": -33.7610042,
        "bearing": 43,
        "road-name": "Pennant Hills Rd"
      }
    }
  ]
}
Schema
{
  "route-id": {
    "type": "String",
    "description": "The TTDS assigned identifier for the route path provided
in the request message. This identifier may be provided in subsequent
Progress requests."
  },
  "system-time": {
    "type": "Number",
    "description": "The wall clock time on the server making the response,
in seconds since epoch."
  },
  "data-time": {
    "type": "Number",
    "description": "The last update time for the data used to compute the
results."
  },
  "travel-time": {
    "type": "Object",
    "description": "The calculated travel time details for the route."
  },
  "properties": {
    "min-seconds": {
      "type": "Number",
      "description": "The minimum expected travel time value, in seconds."
    },
    "max-seconds": {
      "type": "Number",
      "description": "The maximum expected travel time value, in seconds."
    }
  },
  "events": {
    "type": "Array",
    "required": false,
    "description": "A list of currently active events on the specified route
path."
  },
  "properties": {
    "event-id": {
      "type": "Number",
      "description": "An internal ID representing the event that is
persisted across API calls."
    },
    "type": {
      "type": "String",
      "description": "The type of event, currently limited to (rapid)
*deceleration* and (rapid) *acceleration*."
    },
    "detection-time": {
      "type": "Number",
      "description": "The data time in seconds associated with the event.
The event duration can be calculated by *system-time* - *detection-time*."
    },
    "expected-end-time": {
      "type": "Number",
      "description": "The data time in seconds that the event is predicted
to last until. The expected duration can be calculated by *expected-end-
time* - *system-time*."
    },
    "head": {
      "type": "Object",
      "description": "Represents the front of the event, in direction of
travel."
    },
    "properties": {
      "lng": {
        "type": "Number",
        "description": "The longitude value representing the location of the
event. The head of the traffic jam."
      },
      "lat": {
        "type": "Number",
        "description": "The latitude value representing the location of the
event. The head of the traffic jam."
      },
      "bearing": {
        "type": "Number",
        "description": "The bearing in degrees (0<= bearing < 360) at the
head of the event."
```

```
      },
      "road-name": {
        "type": "String",
        "description": "The name of the road at the head of the congestion."
      }
    },
    "tail": {
      "type": "Object",
      "required": false,
      "description": "Represents the back of the event, in direction of
travel. Not provided for (rapid) acceleration events."
    },
    "properties": {
      "lng": {
        "type": "Number",
        "description": "The longitude value representing the location of the
event. The tail of the traffic jam."
      },
      "lat": {
        "type": "Number",
        "description": "The latitude value representing the location of the
event. The tail of the traffic jam."
      }
    },
    "congestion-backlog": {
      "type": "Object",
      "required": false,
      "description": "Details about the congestion backlog, calculated at
the current data time. Not provided for (rapid) acceleration events."
    },
    "properties": {
      "length": {
        "type": "Number",
        "description": "The distance along the road between the head
location and tail location, in metres."
      },
      "min-travel-time": {
        "type": "Number",
        "description": "The minimum expected travel time to travel across
the congestion backlog, in seconds."
      },
      "max-travel-time": {
        "type": "Number",
        "description": "The maximum expected travel time to travel across
the congestion backlog, in seconds."
      }
    }
  }
}
```

- **Response** `400`
- **Response** `401`
- **Response** `500`
- **Response** `503`

- **Response** `504`

Progress API

This request enables mobile applications to regularly report their recent location and optionally receive the remaining travel time on the route and event updates. The route-id returned by a */route* request is provided in the progress request. Routes are cached for use by the progress API and are deleted after 60 minutes of not being referenced. See */route* for more details on route expiration. The progress request must include the cookie returned by the route request.

Progress API calls containing a route-id are using the preferred API.

Routeless progress API calls are an alternative which allow an application developer to contribute position and speed data to improve the overall efficacy of TTDS data. If a route-id is not provided in the request, provide-travel-time and provide-events must be false and the request is providing sample readings only. The routeless progress API is intended for those applications which connect directly from handset devices to the TTDS web server rather than through an application server. Developers using the routeless progress API are encouraged to discuss their requirements with TfNSW. Routeless requests should not provide the Set Cookie header.

Progress Response

POST

/progress

Response Codes

- 200 **OK**: When “route-id” is provided in the request and either “provide-travel-time” or “provide-events” in the request are true, and the mobile app is reporting sample locations on the specified route.
- 202 **Accepted**: When “route-id” is not provided in the request or when “provide-events” and “provide-travel-time” in the request are both false.
- 204 **No Content**: When either “provide-travel-time” or “provide-events” in the request are true, but either a) TTDS cannot find the identified route, or b) the reported samples in the request do not spatially match the route path (device is “off route”).
- 401 **Unauthorized**
- 503 **Service Unavailable**
- 504 **Gateway Timeout**
- **Request**Hide
- Headers

```
Content-Type: application/vnd.ttds-progress+json

Cookie: <value for this route, obtained in response from route request, only
to be provided when route-id is provided>
Body
{
  "route-id": "ab5a61f0-d99f-4769-8c8b-5d6564ef807d",
  "provide-travel-time": true,
  "provide-events": true,
  "samples": [
    {
      "readings": [
        {
          "timestamp": 1399440512,
          "lng": <lng>,
          "lat": <lat>,
          "bearing": 245,
          "speed": 20
        },
        {
          "timestamp": 1399440513,
          "lng": <lng>,
          "lat": <lat>,
          "bearing": 251,
          "speed": 21
        },
        {
          "timestamp": 1399442514,
          "lng": <lng>,
          "lat": <lat>,
          "bearing": 257,
          "speed": 23
        }
      ]
    },
    {
      "readings": [
        {
          "timestamp": 1399440522,
          "lng": <lng>,
          "lat": <lat>,
          "bearing": 259,
          "speed": 53
        },
        {
          "timestamp": 1399440523,
          "lng": <lng>,
          "lat": <lat>,
          "bearing": 259,
          "speed": 53
        },
        {
          "timestamp": 1399440524,
          "lng": <lng>,
          "lat": <lat>,
          "bearing": 260,
          "speed": 54
        }
      ]
    }
  ]
}
```

Schema

```
{
  "route-id": {
    "type": "String",
    "required": false,
    "description": "The TTDS assigned identifier for the route path provided
in response to the Route request. If not supplied, it indicates a routeless
request and updated travel time and updated events are not returned."
  },
  "provide-travel-time": {
    "type": "Boolean",
    "required": false,
    "description": "A flag indicating whether the TTDS should include
predicted travel time along the specified route path in the response
message. Default value: false."
  },
  "provide-events": {
    "type": "Boolean",
    "required": false,
    "description": "A flag indicating whether the TTDS should include events
on the specified route path in the response message. Default value: false."
  },
  "provide-traveltime": {
    "type": "Boolean",
    "required": false,
    "description": "A flag indicating whether the TTDS should include
predicted travel time along the specified route path in the response
message. Default value: false."
  },
  "samples": {
    "type": "Array",
    "description": "The list of samples."
  },
  "properties": {
    "readings": {
      "type": "Array",
      "description": "The list of readings within each sample."
    },
    "properties": {
      "timestamp": {
        "type": "Number",
        "description": "The timestamp when the reading was taken, in seconds
from epoch, integer values."
      },
      "lng": {
        "type": "Number",
        "description": "The longitude value representing the location of the
event. Together with lat must be contained within NSW/ACT."
      },
      "lat": {
        "type": "Number",
        "description": "The latitude value representing the location of the
event. Together with lat must be contained within NSW/ACT."
      },
      "bearing": {
        "type": "Number",
        "required": false,
        "description": "The direction the handset was travelling in at the
time of the reading, in decimal degrees, within the range 0<= bearing < 360,
integernon whole values are rounded."
      },
      "speed": {
        "type": "Number",
        "required": false,
        "description": "The speed the handset was travelling at the time of
the reading, in metres per second, integer valueswithin the range 0<= speed
<= 70, non whole values are rounded."
      }
    }
  }
}
•    Response  200Hide
•    Headers
Cache-Control: private, max-age=60 (set to the internal travel time update
period)

Content-Type: application/vnd.ttds-traveltime+json

Set-Cookie: <See System Behaviour Section, Additional Features Sub-Section,
Cookies Paragraph, ignore in progress response>

X-Server-Hash: 1234
Body
{
  "route-id": "ab5a61f0-d99f-4769-8c8b-5d6564ef807d",
  "system-time": 1399439319,
  "data-time": 1399439319,
  "travel-time": {
    "min-seconds": 336,
    "max-seconds": 400
   }
}
OR
{
  "route-id": "ab5a61f0-d99f-4769-8c8b-5d6564ef807d",
  "system-time": 1399439319,
  "data-time": 1399439319,
  "travel-time": {
    "min-seconds": 336,
    "max-seconds": 400
  },
  "events": [
    {
      "event-id": 123,
      "type": "deceleration",
      "detection-time": 1399123400,
      "expected-end-time": 1399123400,
      "head": {
        "lng": 151.0784132,
        "lat": -33.7629486,
        "bearing": 43,
        "road-name": "M2"
      },
      "tail": {
        "lng": 151.0784132,
        "lat": -33.7629486
      },
      "congestion-backlog": {
        "length": 1000,
        "min-travel-time": 100,
        "max-travel-time": 200
      }
    },
    {
      "event-id": 124,
      "type": "acceleration",
      "detection-time": 1399345600,
      "expected-end-time": 1399345600,
      "head": {
        "lng":151.0485441,
        "lat": -33.7610042,
        "bearing": 43,
        "road-name": "Pennant Hills Rd"
      }
    }
  ]
}
Schema
{
  "route-id": {
    "type": "String",
    "description": "The TTDS assigned identifier for the route path provided
in the request message. This identifier may be provided in subsequent
Progress requests."
  },
  "system-time": {
    "type": "Number",
    "description": "The wall clock time on the server making the response,
in seconds since epoch."
  },
  "data-time": {
    "type": "Number",
    "description": "The last update time for the data used to compute the
results."
  },
  "travel-time": {
    "type": "Object",
    "description": "The calculated travel time details for the route."
  },
  "properties": {
    "min-seconds": {
      "type": "Number",
      "description": "The minimum expected travel time value, in seconds."
    },
    "max-seconds": {
      "type": "Number",
      "description": "The maximum expected travel time value, in seconds."
    }
  },
  "events": {
    "type": "Array",
    "required": false,
    "description": "A list of currently active events on the specified route
path."
  },
  "properties": {
    "event-id": {
      "type": "Number",
      "description": "An internal ID representing the event that is
persisted across API calls."
    },
    "type": {
      "type": "String",
      "description": "The type of event, currently limited to (rapid)
*deceleration* and (rapid) *acceleration*."
    },
    "detection-time": {
      "type": "Number",
      "description": "The data time in seconds associated with the event.
The event duration can be calculated by *system-time* - *detection-time*."
    },
    "expected-end-time": {
      "type": "Number",
      "description": "The data time in seconds that the event is predicted
to last until. The expected duration can be calculated by *expected-end-
time* - *system-time*."
    },
    "head": {
      "type": "Object",
      "description": "Represents the front of the event, in direction of
travel."
    },
    "properties": {
      "lng": {
        "type": "Number",
        "description": "The longitude value representing the location of the
event. The head of the traffic jam."
      },
      "lat": {
        "type": "Number",
        "description": "The latitude value representing the location of the
event. The head of the traffic jam."
      },
      "bearing": {
        "type": "Number",
        "description": "The bearing in degrees (0<= bearing < 360) at the
head of the event."
      },
      "road-name": {
        "type": "String",
        "description": "The name of the road at the head of the congestion."
      }
    },
    "tail": {
      "type": "Object",
      "required": false,
      "description": "Represents the back of the event, in direction of
travel. Not provided for (rapid) acceleration events."
```

```
    },
    "properties": {
      "lng": {
        "type": "Number",
        "description": "The longitude value representing the location of the
event. The tail of the traffic jam."
      },
      "lat": {
        "type": "Number",
        "description": "The latitude value representing the location of the
event. The tail of the traffic jam."
      }
    },
    "congestion-backlog": {
      "type": "Object",
      "required": false,
      "description": "Details about the congestion backlog, calculated at
the current data time. Not provided for (rapid) acceleration events."
    },
    "properties": {
      "length": {
        "type": "Number",
        "description": "The distance along the road between the head
location and tail location, in metres."
      },
      "min-travel-time": {
        "type": "Number",
        "description": "The minimum expected travel time to travel across
the congestion backlog, in seconds."
      },
      "max-travel-time": {
        "type": "Number",
        "description": "The maximum expected travel time to travel across
the congestion backlog, in seconds."
      }
    }
  }
}
```

- **Response `202`**
- **Response `204`**
- **Response `401`**
- **Response `503`**
- **Response `504`**

Events API

This request enables a mobile application to request the collection of all current events across the entire road network.

Events Request

GET

/events

Response Codes

- 200 **OK**
- 304 **Not Modified**
- 401 **Unauthorized**
- 503 **Service Unavailable**
- 504 **Gateway Timeout**
- **Request**Hide
- Headers
- If-Modified-Since: <HTTP-datetime>
- **Response** 200Hide
- Headers

```
Content-Type: application/vnd.ttds-traveltime+json
Body
{
  "system-time": 1399439319,
  "data-time": 1399439319,
  "events": [
    {
      "event-id": 123,
      "type": "deceleration",
      "detection-time": 1399123400,
      "expected-end-time": 1399123400,
      "head": {
        "lng": 151.0784132,
        "lat": -33.7629486,
        "bearing": 43,
        "road-name": "M2"
      },
      "tail": {
        "lng": 151.0784132,
        "lat": -33.7629486
      },
      "congestion-backlog": {
        "length": 1000,
        "min-travel-time": 100,
        "max-travel-time": 200
      }
    },
    {
      "event-id": 124,
      "type": "acceleration",
      "detection-time": 1399345600,
      "expected-end-time": 1399345600,
      "head": {
        "lng":151.0485441,
        "lat": -33.7610042,
        "bearing": 43,
        "road-name": "Pennant Hills Rd"
      }
    }
  ]
}
Schema
{
  "system-time": {
    "type": "Number",
    "description": "The wall clock time on the server making the response, 
in seconds since epoch."
  },
  "data-time": {
    "type": "Number",
    "description": "The last updated time for the data used to compute the 
results."
  },
  "events": {
    "type": "Array",
    "required": false,
    "description": "A list of currently active events on the specified route 
path."
  },
  "properties": {
    "event-id": {
      "type": "Number",
      "description": "An internal ID representing the event that is 
persisted across API calls."
    },
    "type": {
      "type": "String",
      "description": "The type of event, currently limited to (rapid) 
*deceleration* and (rapid) *acceleration*."
    },
    "detection-time": {
      "type": "Number",
      "description": "The data time in seconds associated with the event. 
The event duration can be calculated by *system-time* - *detection-time*."
    },
    "expected-end-time": {
      "type": "Number",
      "description": "The data time in seconds that the event is predicted 
to last until. The expected duration can be calculated by *expected-end-
time* - *system-time*."
    },
    "head": {
      "type": "Object",
      "description": "Represents the front of the event, in direction of 
travel."
    },
    "properties": {
      "lng": {
        "type": "Number",
        "description": "The longitude value representing the location of the 
event. The head of the traffic jam."
      },
      "lat": {
        "type": "Number",
        "description": "The latitude value representing the location of the
event. The head of the traffic jam."
      },
      "bearing": {
        "type": "Number",
        "description": "The bearing in degrees (0<= bearing < 360) at the
head of the event."
      },
      "road-name": {
        "type": "String",
        "description": "The name of the road at the head of the congestion."
      }
    },
    "tail": {
      "type": "Object",
      "required": false,
      "description": "Represents the back of the event, in direction of
travel. Not provided for (rapid) acceleration events."
    },
    "properties": {
      "lng": {
        "type": "Number",
        "description": "The longitude value representing the location of the
event. The tail of the traffic jam."
      },
      "lat": {
        "type": "Number",
        "description": "The latitude value representing the location of the
event. The tail of the traffic jam."
      }
    },
    "congestion-backlog": {
      "type": "Object",
      "required": false,
      "description": "Details about the congestion backlog, calculated at
the current data time. Not provided for (rapid) acceleration events."
    },
    "properties": {
      "length": {
        "type": "Number",
        "description": "The distance along the road between the head
location and tail location, in metres."
      },
      "min-travel-time": {
        "type": "Number",
        "description": "The minimum expected travel time to travel across
the congestion backlog, in seconds."
      },
      "max-travel-time": {
        "type": "Number",
        "description": "The maximum expected travel time to travel across
the congestion backlog, in seconds."
      }
    }
  }
```

```
}
```

- **Response 304**
- **Response 401**
- **Response 503**
- **Response 504**

API Validation Rules

API validation rules are defined below.

All requests

- Only GET or POST methods are supported (GET /events, POST /route, POST /progress etc). All other methods are prohibited (405 / METHOD NOT ALLOWED)

GET requests

/events

- If-Modified-Since header (if present) must be in EEE, dd MMM yyyy HH:mm:ss GMT format (400 / BAD REQUEST)
- If-Modified-since time (if present) must be before the last time events were cached (304 / NOT MODIFIED)

> NOTE: errors are only signalled if the request actually includes an If-Modified-Since header. A badly behaved client can make repeated requests without error.

POST Requests

All POST requests:

- Must supply a User-Agent header with a value in the form <appname> / <majorversion>.<minorversion>
  - The java regex to verify this is: "([^\\/]*)\\/\\s*([A-Za-z0-9]*)\\s*.\\s*([A-Za-z0-9]*)\\s*".
- Request message must be supplied (400 / BAD REQUEST)
- The request message must be valid JSON (400 / BAD REQUEST)

/route

- Content-Type header must be application/vnd.ttds-route+json (400 / BAD REQUEST)
- The message must contain at least one valid polyline section in encoded-paths, and no null or invalid sections (400 / BAD REQUEST)
- Polyline section lengths must each be <= 16000 characters (400 / BAD REQUEST)
- All points in each polyline must be within a defined latitude / longitude bounding box (400 / BAD REQUEST). The box is defined in ttdsapi.properties as all of Australia including Tasmania:
  - NE -9.228820, 159.278717

  - SW -54.640301, 112.921112
- If an arrival or departure time is specified, it must be in the range 1 July 2014 to 1 July 2024 Australian EST (400 / BAD REQUEST)
- If route mapping fails, an error is returned (400 / BAD REQUEST). This can happen particularly if start or end point is not close enough to a valid road element.
- If no response is received after a given timeout period an error is returned (500 / INTERNAL SERVICE ERROR)

/progress

- Content-Type header must be application/vnd.ttds-progress+json (400 / BAD REQUEST)
- If route-id is supplied, it must be 36 characters in length (400 / BAD REQUEST)
- If route-id is not supplied, provide-events and provide-traveltime may not be true (400 / BAD REQUEST)
- If route-id is not supplied, samples must be supplied (400 / BAD REQUEST)
- If samples are supplied, they must not be null, and must contain valid reading elements (400 / BAD REQUEST)
- reading elements may not be null; each component must be valid (400 / BAD REQUEST)
  - speed must be in the range 0-70 m/s (approx 200 kph / 125 mph)
  - bearing must be in the range 0-360 degrees.
  - lat and lng must be in the configured bounding box (see /route above).