V-DAQ Integrations API

Version 2.0.1

An API for getting vehicle metadata, trip data, events, live telemetry & timeseries data from the V-DAQ platform.

`https://api.v-daq.com.au`

Created 21 October 2026

CONFIDENTIAL - For authorised integrators only

Authentication

The API uses a scoped API key which can be generated on the platform site, under the apps section.

Once an API key has been defined with its scopes, it is used in the header of each request to the integrations API:

```
X-API-Key: <your-api-key>
```

If the API key is missing or invalid, the API returns a 403 Forbidden response.

Domain

```
https://api.v-daq.com.au
```

All endpoint paths shown in this document are relative to this domain.

Error Responses

All endpoints return errors in a consistent format:

```
{
  "error": "An error occurred"
}
```

Status Code	Meaning
400	Bad Request - invalid parameters or missing required fields
403	Forbidden - API key lacks permission for the requested resource
404	Not Found - resource does not exist or is not accessible
429	Too Many Requests - rate limit exceeded (see Rate Limits section)
500	Internal Server Error - unexpected server-side failure

When the rate limit is exceeded, the API returns a 429 status code with the following body:

```
{
  "error": "rate limit exceeded"
}
```

Usage Notes

Time Range Limits

Endpoints that accept startTime and endTime query parameters require both values in ISO 8601 format (RFC 3339). The following constraints apply:

Constraint	Detail
Format	ISO 8601 / RFC 3339 (e.g. 2025-01-01T00:00:00Z)
Ordering	The startTime must be before the endTime
Required	Both startTime and endTime must be provided

Maximum Time Range by Endpoint

Each endpoint group has its own maximum allowed duration between startTime and endTime. Requests exceeding the limit will receive a 400 Bad Request response.

Endpoints	Maximum Range
/api/v1/trip/*	48 hours
/api/v1/event/*	24 hours
/api/v1/location/*	48 hours
/api/v1/mass/*	48 hours

Rate Limits

API requests are subject to per-API-key rate limiting. Each API key has its own independent request quota. Exceeding the limit will result in a 429 Too Many Requests response.

Limits

Each API key is allowed up to 60 requests per 30-second window (equivalent to 2 requests/second sustained). The window is a fixed interval: once the window elapses, the counter resets automatically.

Setting	Value
Sustained rate	2 req/s per API key
Window size	30 seconds
Max requests per window	60

Rate Limit Response Headers

Every API response includes the following headers to help you track your usage and avoid hitting the limit:

Header	Description
`X-RateLimit-Limit`	Maximum number of requests allowed per window (60)
`X-RateLimit-Remaining`	Number of requests remaining in the current window
`X-RateLimit-Reset`	Unix epoch timestamp (seconds) when the current window resets

If you receive a 429 response, wait until the time indicated by X-RateLimit-Reset before retrying.

Vehicles

GET /api/v1/vehicle

Get all vehicles

Retrieves all vehicle metadata scoped to the authenticated API key

Headers

Name	Required	Description
X-API-Key	Yes	API Key

Example Response (200)


```
{
  "data": [
    {
      "account": "27ea549ffb505fcfbac7b7208131a7c2",
      "driver": {
        "driverName": "Tom",
        "employeeID": "EMP123"
      },
      "engineTime": 5000,
      "labels": [
        {
          "name": "color",
          "value": "blue"
        }
      ],
      "odometer": 150000,
      "regState": "NSW",
      "registration": "ABC123",
      "vehicle": "4b5bac552b73526ebaf6010574c39b76",
      "vin": "1HGCM82633A123456"
    }
  ]
}
```

VehicleMetadata Fields (each item in data array)

Field	Type	Description
`account`	string	Account-scoped vehicle identifier
`driver`	DriverSummary	Currently assigned driver
`engineTime`	number	Total engine run time (hours)
`labels`	[]Label	Custom labels attached to the vehicle
`odometer`	number	Current odometer reading (km)
`regState`	string	Registration state
`registration`	string	Vehicle registration
`vehicle`	string	Vehicle entity ID
`vin`	string	Vehicle Identification Number

DriverSummary Fields

Field	Type	Description
`driverName`	string	Driver name
`employeeID`	string	Employee ID

Label Fields

Field	Type	Description
`name`	string	Label name
`value`	string	Label value

Vehicles

GET /api/v1/vehicle/{vehicleID}

Get vehicle by ID

Retrieves a single vehicle's metadata by ID, scoped to the authenticated API key

Headers

Name	Required	Description
X-API-Key	Yes	API Key

Path Parameters

Name	Required	Description
vehicleID	Yes	Vehicle ID

Example Response (200)


```
{
  "data": {
    "account": "27ea549ffb505fcfbac7b7208131a7c2",
    "driver": {
      "driverName": "Tom",
      "employeeID": "EMP123"
    },
    "engineTime": 5000,
    "labels": [
      {
        "name": "color",
        "value": "blue"
      }
    ],
    "odometer": 150000,
    "regState": "NSW",
    "registration": "ABC123",
    "vehicle": "4b5bac552b73526ebaf6010574c39b76",
    "vin": "1HGCM82633A123456"
  }
}
```

VehicleMetadata Fields

Field	Type	Description
`account`	string	Account-scoped vehicle identifier
`driver`	DriverSummary	Currently assigned driver
`engineTime`	number	Total engine run time (hours)
`labels`	[]Label	Custom labels attached to the vehicle
`odometer`	number	Current odometer reading (km)
`regState`	string	Registration state
`registration`	string	Vehicle registration
`vehicle`	string	Vehicle entity ID
`vin`	string	Vehicle Identification Number

DriverSummary Fields

Field	Type	Description
`driverName`	string	Driver name
`employeeID`	string	Employee ID

Label Fields

Field	Type	Description
`name`	string	Label name
`value`	string	Label value

Trips

GET `/api/v1/trip`

Get all trips for an account

Retrieves all trips for all vehicles in the account over a time range of up to 48 hours

> *Note: The trip start time must fall in between the start and end time to be included in the query.*

Headers

Name	Required	Description
`X-API-Key`	Yes	API Key

Query Parameters

Name	Required	Type	Description
`startTime`	Yes	string	Start time (ISO 8601 format)
`endTime`	Yes	string	End time (ISO 8601 format)

Example Response (200)


```
{
  "data": [
    {
      "ID": "9c23208d5b164f51be2b824322be532e",
      "account": "e31c86946fff4fb88fe5ad57970e000d",
      "created": "2026-02-04T15:07:00Z",
      "distance": 219.12,
      "driver": {
        "driverName": "Tom",
        "employeeID": "EMP123"
      },
      "duration": 342353,
      "end": {
        "latitude": -33.8643964,
        "location": "Flemington, Homebush West NSW 2140, Australia",
        "longitude": 151.0706899,
        "odometer": 134124,
        "time": "2026-02-04T14:46:13Z"
      },
      "idleDuration": 2324,
      "start": {
        "latitude": -33.8643964,
        "location": "Flemington, Homebush West NSW 2140, Australia",
        "longitude": 151.0706899,
        "odometer": 134124,
        "time": "2026-02-04T14:46:13Z"
      },
      "vehicle": "4a2ffe0735634bbf837d5bc243ca998c"
    }
  ]
}
```


TripSummary Fields (each item in data array)

Field	Type	Description
`ID`	string	Trip identifier
`account`	string	Account identifier
`created`	string	Trip creation timestamp
`distance`	number	Trip distance (km)
`driver`	DriverSummary	Driver assigned during trip
`duration`	integer	Trip duration (seconds)
`end`	TripLocationSummary	Trip end details
`idleDuration`	integer	Idle time during trip (seconds)
`start`	TripLocationSummary	Trip start details
`vehicle`	string	Vehicle entity ID

DriverSummary Fields

Field	Type	Description
`driverName`	string	Driver name
`employeeID`	string	Employee ID

TripLocationSummary Fields

Field	Type	Description
`latitude`	number	Latitude (decimal degrees)
`location`	string	Reverse-geocoded address
`longitude`	number	Longitude (decimal degrees)
`odometer`	number	Odometer reading at this point (km)
`time`	string	Timestamp (ISO 8601)

Trips

GET `/api/v1/trip/{vehicleID}`

Get trips for a vehicle

Retrieves trips for a specific vehicle over a time range of up to 48 hours

> *Note: The trip start time must fall in between the start and end time to be included in the query.*

Headers

Name	Required	Description
`X-API-Key`	Yes	API Key

Path Parameters

Name	Required	Description
`vehicleID`	Yes	Vehicle ID

Query Parameters

Name	Required	Type	Description
`startTime`	Yes	string	Start time (ISO 8601 format)
`endTime`	Yes	string	End time (ISO 8601 format)

Example Response (200)


```
{
  "data": [
    {
      "ID": "9c23208d5b164f51be2b824322be532e",
      "account": "e31c86946fff4fb88fe5ad57970e000d",
      "created": "2026-02-04T15:07:00Z",
      "distance": 219.12,
      "driver": {
        "driverName": "Tom",
        "employeeID": "EMP123"
      },
      "duration": 342353,
      "end": {
        "latitude": -33.8643964,
        "location": "Flemington, Homebush West NSW 2140, Australia",
        "longitude": 151.0706899,
        "odometer": 134124,
        "time": "2026-02-04T14:46:13Z"
      },
      "idleDuration": 2324,
      "start": {
        "latitude": -33.8643964,
        "location": "Flemington, Homebush West NSW 2140, Australia",
        "longitude": 151.0706899,
        "odometer": 134124,
        "time": "2026-02-04T14:46:13Z"
      },
      "vehicle": "4a2ffe0735634bbf837d5bc243ca998c"
    }
  ]
}
```


TripSummary Fields (each item in data array)

Field	Type	Description
`ID`	string	Trip identifier
`account`	string	Account identifier
`created`	string	Trip creation timestamp
`distance`	number	Trip distance (km)
`driver`	DriverSummary	Driver assigned during trip
`duration`	integer	Trip duration (seconds)
`end`	TripLocationSummary	Trip end details
`idleDuration`	integer	Idle time during trip (seconds)
`start`	TripLocationSummary	Trip start details
`vehicle`	string	Vehicle entity ID

DriverSummary Fields

Field	Type	Description
`driverName`	string	Driver name
`employeeID`	string	Employee ID

TripLocationSummary Fields

Field	Type	Description
`latitude`	number	Latitude (decimal degrees)
`location`	string	Reverse-geocoded address
`longitude`	number	Longitude (decimal degrees)
`odometer`	number	Odometer reading at this point (km)
`time`	string	Timestamp (ISO 8601)

Events

GET `/api/v1/event`

Get all events for an account

Retrieves all events for all vehicles in the account within a specified time range

> *Note: Event endpoints have a maximum time range of 24 hours.*

Event Types

Value	Description
`HARSH_CORNERING`	Harsh cornering detected (lateral acceleration on the Y-axis exceeds threshold, in m/s^2)
`HARSH_BRAKING`	Harsh braking detected (deceleration on the X-axis exceeds threshold, in m/s^2)
`HARSH_ACCELERATION`	Harsh acceleration detected (acceleration on the X-axis exceeds threshold, in m/s^2)
`OVERSPEED`	Overspeed detected (vehicle exceeds speed threshold, in m/s)

Response Fields by Event Type

The presence of the following fields varies by event type. Common fields (ID, eventType, name, vehicle, driver, location, config, startTime, endTime, time, updated) are always present.

Event Type	duration	maxSpeed	maxAcceleration	direction	distance
`OVERSPEED`	✓	✓	✗	✗	✓
`HARSH_BRAKING`	✓	✓	✓	✗	✓
`HARSH_ACCELERATION`	✓	✓	✓	✗	✗
`HARSH_CORNERING`	✓	✓	✓	✓	✓

Headers

Name	Required	Description
`X-API-Key`	Yes	API Key

Query Parameters

Name	Required	Type	Description
`startTime`	Yes	string	Start time (ISO 8601 format)
`endTime`	Yes	string	End time (ISO 8601 format)
`eventTypes`		string	Comma-separated event types filter (e.g. OVERSPEED,HARSH_BRAKING)

Example Response (200)


```
{
  "data": [
    {
      "ID": "2cc1356189a14a11a0c62e60adaa36da",
      "config": {
        "duration": 3,
        "threshold": 2.5
      },
      "direction": "LEFT",
      "distance": 24.86049938205,
      "driver": {
        "driverName": "Tom",
        "employeeID": "EMP123"
      },
      "duration": 9.23,
      "endTime": "2026-02-05T02:15:15+00:00",
      "eventType": "HARSH_CORNERING",
      "location": {
        "end": {
          "address": "George Tyson Drive, Canberra 2609",
          "latitude": -35.3096471,
          "longitude": 149.1897873
        },
        "start": {
          "address": "George Tyson Drive, Canberra 2609",
          "latitude": -35.3096471,
          "longitude": 149.1897873
        }
      },
      "maxAcceleration": 3.4400000572,
      "maxSpeed": 8.7919998169,
      "name": "TEST HARSH CORNERING",
      "startTime": "2026-02-05T02:15:14+00:00",
      "time": "2026-02-05T02:15:15+00:00",
      "updated": "2026-02-05T02:15:15+00:00",
      "vehicle": "a6d3ff0dbde74fc6ac419d65111a3df5"
    }
  ]
}
```

EventSummary Fields (each item in data array)

Field	Type	Description
`ID`	string	Event identifier
`config`	EventConfig	Event detection configuration
`direction`	string	Direction (for cornering events, e.g. LEFT, RIGHT)
`distance`	number	Distance covered during event (metres)
`driver`	DriverSummary	Driver assigned at time of event
`duration`	number	Event duration (seconds)
`endTime`	string	Event end time
`eventType`	EventType	One of the event type enum values
`location`	EventLocationPair	Start and end location pair
`maxAcceleration`	number	Max acceleration magnitude during event (m/s^2, always positive; omitted when N/A)
`maxSpeed`	number	Max speed during event (m/s; omitted when N/A)
`name`	string	Event display name
`startTime`	string	Event start time
`time`	string	Event timestamp
`updated`	string	Last updated timestamp
`vehicle`	string	Vehicle entity ID

EventConfig Fields

Field	Type	Description
`duration`	number	Minimum duration for trigger (seconds)
`threshold`	number	Trigger threshold: speed (m/s) for OVERSPEED, acceleration (m/s^2) for other events

DriverSummary Fields

Field	Type	Description
`driverName`	string	Driver name
`employeeID`	string	Employee ID

EventLocationPair Fields

Field	Type	Description
`end`	EventLocation	End location
`start`	EventLocation	Start location

EventLocation Fields

Field	Type	Description
`address`	string	Reverse-geocoded address
`latitude`	number	Latitude (decimal degrees)
`longitude`	number	Longitude (decimal degrees)

Events

GET /api/v1/event/{vehicleID}

Get events for a vehicle

Retrieves events for a specific vehicle within a specified time range

> *Note: Event endpoints have a maximum time range of 24 hours.*

Event Types

Value	Description
HARSH_CORNERING	Harsh cornering detected (lateral acceleration on the Y-axis exceeds threshold, in m/s^2)
HARSH_BRAKING	Harsh braking detected (deceleration on the X-axis exceeds threshold, in m/s^2)
HARSH_ACCELERATION	Harsh acceleration detected (acceleration on the X-axis exceeds threshold, in m/s^2)
OVERSPEED	Overspeed detected (vehicle exceeds speed threshold, in m/s)

Response Fields by Event Type

The presence of the following fields varies by event type. Common fields (ID, eventType, name, vehicle, driver, location, config, startTime, endTime, time, updated) are always present.

Event Type	duration	maxSpeed	maxAcceleration	direction	distance
OVERSPEED	✓	✓	✗	✗	✓
HARSH_BRAKING	✓	✓	✓	✗	✓
HARSH_ACCELERATION	✓	✓	✓	✗	✗
HARSH_CORNERING	✓	✓	✓	✓	✓

Headers

Name	Required	Description
X-API-Key	Yes	API Key

Path Parameters

Name	Required	Description
vehicleID	Yes	Vehicle ID

Query Parameters

Name	Required	Type	Description
`startTime`	Yes	string	Start time (ISO 8601 format)
`endTime`	Yes	string	End time (ISO 8601 format)
`eventTypes`		string	Comma-separated event types filter (e.g. OVERSPEED,HARSH_BRAKING)

Example Response (200)


```
{
  "data": [
    {
      "ID": "2cc1356189a14a11a0c62e60adaa36da",
      "config": {
        "duration": 3,
        "threshold": 2.5
      },
      "direction": "LEFT",
      "distance": 24.86049938205,
      "driver": {
        "driverName": "Tom",
        "employeeID": "EMP123"
      },
      "duration": 9.23,
      "endTime": "2026-02-05T02:15:15+00:00",
      "eventType": "HARSH_CORNERING",
      "location": {
        "end": {
          "address": "George Tyson Drive, Canberra 2609",
          "latitude": -35.3096471,
          "longitude": 149.1897873
        },
        "start": {
          "address": "George Tyson Drive, Canberra 2609",
          "latitude": -35.3096471,
          "longitude": 149.1897873
        }
      },
      "maxAcceleration": 3.4400000572,
      "maxSpeed": 8.7919998169,
      "name": "TEST HARSH CORNERING",
      "startTime": "2026-02-05T02:15:14+00:00",
      "time": "2026-02-05T02:15:15+00:00",
      "updated": "2026-02-05T02:15:15+00:00",
      "vehicle": "a6d3ff0dbde74fc6ac419d65111a3df5"
    }
  ]
}
```

EventSummary Fields (each item in data array)

Field	Type	Description
`ID`	string	Event identifier
`config`	EventConfig	Event detection configuration
`direction`	string	Direction (for cornering events, e.g. LEFT, RIGHT)
`distance`	number	Distance covered during event (metres)
`driver`	DriverSummary	Driver assigned at time of event
`duration`	number	Event duration (seconds)
`endTime`	string	Event end time
`eventType`	EventType	One of the event type enum values
`location`	EventLocationPair	Start and end location pair
`maxAcceleration`	number	Max acceleration magnitude during event (m/s^2, always positive; omitted when N/A)
`maxSpeed`	number	Max speed during event (m/s; omitted when N/A)
`name`	string	Event display name
`startTime`	string	Event start time
`time`	string	Event timestamp
`updated`	string	Last updated timestamp
`vehicle`	string	Vehicle entity ID

EventConfig Fields

Field	Type	Description
`duration`	number	Minimum duration for trigger (seconds)
`threshold`	number	Trigger threshold: speed (m/s) for OVERSPEED, acceleration (m/s^2) for other events

DriverSummary Fields

Field	Type	Description
`driverName`	string	Driver name
`employeeID`	string	Employee ID

EventLocationPair Fields

Field	Type	Description
`end`	EventLocation	End location
`start`	EventLocation	Start location

EventLocation Fields

Field	Type	Description
`address`	string	Reverse-geocoded address
`latitude`	number	Latitude (decimal degrees)
`longitude`	number	Longitude (decimal degrees)

Location

GET /api/v1/location/{vehicleID}

Export location data for a vehicle

Exports a vehicle's location data over a time range of up to 48 hours. Returns a presigned URL to a gzipped JSON array of location records. Each record contains latitude, longitude, altitude, accuracy metrics, satellite count, fix type, PDOP, and quality score. Timestamps are in RFC 3339 UTC format.

> *Note: The location export returns a gzipped JSON array of location records. Download and decompress the file at the presigned URL to access the data.*

> *Note: Location endpoints have a maximum time range of 48 hours.*

GNSS Fix Types

The fix_type field in each location record indicates the type of GNSS solution used.

Value	Description
`GNSS_AND_UDR`	GNSS augmented with dead-reckoning (best accuracy, most common)
`GNSS_3D`	3D GNSS fix (latitude, longitude, altitude)
`GNSS_2D`	2D GNSS fix (latitude, longitude only)
`NO_FIX`	No GNSS fix available at this time
`DEAD_RECKONING`	Dead-reckoning only (no GNSS fix)

Headers

Name	Required	Description
`X-API-Key`	Yes	API Key

Path Parameters

Name	Required	Description
`vehicleID`	Yes	Vehicle ID

Query Parameters

Name	Required	Type	Description
`startTime`	Yes	string	Start time (ISO 8601 format)
`endTime`	Yes	string	End time (ISO 8601 format)

Example Response (200)


```
{
  "expires_at": "2026-01-29T11:00:00Z",
  "url": "https://example.com/presigned-url"
}
```

Location Record Example

The file at the presigned URL is a gzipped JSON array of location records. Each record is one GNSS fix:


```
[
  {
    "source": "5161ce282d8e414eb332d67d5cc81504",
    "time": "2026-09-15T04:04:24Z",
    "updated": "2026-09-15T04:05:40Z",
    "received": "2026-09-15T04:04:27Z",
    "latitude": -27.746608,
    "longitude": 153.2341662,
    "altitude": 54.552,
    "horizontal_accuracy": 1.265,
    "satellite_number": 14,
    "fix_type": "GNSS_AND_UDR",
    "pdop": 1.36,
    "quality": 0.94
  },
  {
    "source": "5161ce282d8e414eb332d67d5cc81504",
    "time": "2026-09-15T04:15:23Z",
    "updated": "2026-09-15T04:15:40Z",
    "received": "2026-09-15T04:15:23Z",
    "latitude": -27.8854056,
    "longitude": 153.3156535,
    "altitude": 40.489,
    "horizontal_accuracy": 1.36,
    "satellite_number": 0,
    "fix_type": "UDR_ONLY",
    "pdop": 99.99,
    "quality": 0.93
  }
]
```

Mass

GET /api/v1/mass/{vehicleID}

Export mass data for a vehicle

Exports a vehicle's onboard mass records over a time range of up to 48 hours, with the gross and per-axle weights of each weighing collapsed into one record. Each record contains a timestamp, gross weight, and a list of individual axle weights with their order positions and addresses. Returns a presigned URL to a gzipped JSON array of mass records.

> *Note: The mass export returns a gzipped JSON array of weight records. Download and decompress the file at the presigned URL to access the data.*

> *Note: Mass endpoints have a maximum time range of 48 hours.*

Mass Sensor Types

The sensor field in each mass record indicates the type of weighing. A weighing event produces one GROSS record and multiple AXLE records.

Value	Description
`GROSS`	Whole-vehicle gross weight reading (includes all axles)
`AXLE`	Individual axle weight reading (positional order in the vehicle)

Headers

Name	Required	Description
`X-API-Key`	Yes	API Key

Path Parameters

Name	Required	Description
`vehicleID`	Yes	Vehicle ID

Query Parameters

Name	Required	Type	Description
`startTime`	Yes	string	Start time (ISO 8601 format)
`endTime`	Yes	string	End time (ISO 8601 format)

Example Response (200)

```
{
  "expires_at": "2026-01-29T11:00:00Z",
  "url": "https://example.com/presigned-url"
}
```

Mass Record Example

The file at the presigned URL is a gzipped JSON array of mass records. Each record is one weighing, with the gross weight and per-axle weights collapsed together:


```
[
  {
    "source": "9376149b8bfd4087b86645356050c5d9",
    "timestamp": "2026-09-15T04:09:27Z",
    "quality": 1,
    "gross": {
      "weight": 35910,
      "address": "70AC084BCD2A"
    },
    "axles": [
      {
        "order": 1,
        "weight": 6400,
        "address": "70AC084BCD2A"
      },
      {
        "order": 2,
        "weight": 4360,
        "address": "70AC084BCD2A"
      },
      {
        "order": 3,
        "weight": 7650,
        "address": "385B4483EE9B"
      },
      {
        "order": 4,
        "weight": 17500,
        "address": "385B4483EE9B"
      }
    ]
  }
]
```

Live

GET `/api/v1/live`

Get live data for all vehicles

Retrieves the most recent live telemetry data (parked state, location, mass) for every vehicle the authenticated API key is permitted to access via the live ACL scope.

> *Note: Returns live data for every vehicle the API key can access.*

Live Data Response Structure

Each vehicle object contains a vehicle identifier and, when known, whether the vehicle is parked. It also contains a location and/or mass object when the vehicle has reported them recently.

Vehicle fields:

- vehicle - Vehicle identifier
- parked - true when the vehicle is parked, false when it is not; omitted when the platform has no parked state for it

Location fields:

- lat, lng - GNSS position in decimal degrees (WGS84)
- heading - Direction of travel in degrees from north (0-360)
- alt - Altitude above sea level in metres
- speed - Ground speed in km/h
- timestamp - Time of the GNSS fix (RFC 3339 UTC)
- received - Time the platform received the fix (RFC 3339 UTC)

Mass fields:

- gross - Total gross vehicle weight in kg
- axles - Array of per-axle weight readings
- unit - Unit of measurement (kg)
- received - Time the platform received the mass reading (RFC 3339 UTC)
- timestamp - Time of the mass reading (RFC 3339 UTC)

Headers

Name	Required	Description
`X-API-Key`	Yes	API Key

Example Response (200)


```
{
  "data": [
    {
      "location": {
        "alt": 80.73,
        "heading": 268.22,
        "lat": -33.8010346,
        "lng": 150.8644342,
        "received": "2026-08-07T00:00:00Z",
        "speed": 0,
        "timestamp": "2026-08-07T00:00:00Z"
      },
      "mass": {
        "axles": [
          {
            "axle": "STEER",
            "mass": 6100
          }
        ],
        "gross": 49130,
        "received": "2026-08-07T00:00:00Z",
        "timestamp": "2026-08-07T00:00:00Z",
        "unit": "kg"
      },
      "parked": false,
      "vehicle": "04ab96522b1346ea9a925ba9eccf5db3"
    }
  ]
}
```


LiveData Fields (each item in data array)

Field	Type	Description
`location`	LiveLocation	Current location data, if available
`mass`	LiveMass	Current onboard mass readings, if available
`parked`	boolean	Whether the vehicle is currently parked, if the platform has a parked state for it
`vehicle`	string	Vehicle entity ID

LiveLocation Fields

Field	Type	Description
`alt`	number	Altitude above sea level (metres)
`heading`	number	Heading in degrees from north (0-360)
`lat`	number	Latitude (decimal degrees)
`lng`	number	Longitude (decimal degrees)
`received`	string	Time the location record was received by the platform (RFC 3339 UTC)
`speed`	number	Ground speed (km/h)
`timestamp`	string	Time of the Location record (RFC 3339 UTC)

LiveMass Fields

Field	Type	Description
`axles`	[]LiveAxle	Per-axle mass readings
`gross`	number	Total gross vehicle weight (kg)
`received`	string	Time the mass reading was received by the platform (RFC 3339 UTC)
`timestamp`	string	Time of the mass reading (RFC 3339 UTC)
`unit`	string	Unit of measurement for mass values

LiveAxle Fields

Field	Type	Description
`axle`	string	Axle identifier (e.g. STEER, DRIVE, AXLE1, AXLE2 etc.)
`mass`	number	Mass on this axle (kg)

Live

GET /api/v1/live/{vehicleID}

Get live data for a vehicle

Retrieves the most recent live telemetry data for a specific vehicle, including whether it is parked, current GNSS position (latitude, longitude, heading, altitude, speed) and onboard mass readings (gross weight, per-axle weights).

> *Note: Returns live data for a single vehicle.*

Live Data Response Structure

Each vehicle object contains a vehicle identifier and, when known, whether the vehicle is parked. It also contains a location and/or mass object when the vehicle has reported them recently.

Vehicle fields:

- vehicle - Vehicle identifier
- parked - true when the vehicle is parked, false when it is not; omitted when the platform has no parked state for it

Location fields:

- lat, lng - GNSS position in decimal degrees (WGS84)
- heading - Direction of travel in degrees from north (0-360)
- alt - Altitude above sea level in metres
- speed - Ground speed in km/h
- timestamp - Time of the GNSS fix (RFC 3339 UTC)
- received - Time the platform received the fix (RFC 3339 UTC)

Mass fields:

- gross - Total gross vehicle weight in kg
- axles - Array of per-axle weight readings
- unit - Unit of measurement (kg)
- received - Time the platform received the mass reading (RFC 3339 UTC)
- timestamp - Time of the mass reading (RFC 3339 UTC)

Headers

Name	Required	Description
`X-API-Key`	Yes	API Key

Path Parameters

Name	Required	Description
`vehicleID`	Yes	Vehicle ID

Example Response (200)


```
{
  "location": {
    "alt": 80.73,
    "heading": 268.22,
    "lat": -33.8010346,
    "lng": 150.8644342,
    "received": "2026-08-07T00:00:00Z",
    "speed": 0,
    "timestamp": "2026-08-07T00:00:00Z"
  },
  "mass": {
    "axles": [
      {
        "axle": "STEER",
        "mass": 6100
      }
    ],
    "gross": 49130,
    "received": "2026-08-07T00:00:00Z",
    "timestamp": "2026-08-07T00:00:00Z",
    "unit": "kg"
  },
  "parked": false,
  "vehicle": "04ab96522b1346ea9a925ba9eccf5db3"
}
```

Assets

GET /api/v1/assets

Get all assets

Retrieves all assets scoped to the authenticated API key, including the latest odometer reading carried on each asset row

Headers

Name	Required	Description
X-API-Key	Yes	API Key

Example Response (200)


```
{
  "data": [
    {
      "ID": "27ea549ffb505fcfbac7b7208131a7c2",
      "accountActive": true,
      "assetName": "Example Asset",
      "assetType": "trailer",
      "data": {
        "location": {
          "alt": 12.3,
          "heading": 90,
          "lat": -33.8688,
          "lng": 151.2093,
          "received": "2026-09-29T00:36:53Z",
          "speed": 43.2,
          "timestamp": "2026-09-29T00:35:47Z"
        },
        "odometer": {
          "odometer": 27000,
          "received": "2026-09-29T00:36:53Z",
          "timestamp": "2026-09-29T00:35:47Z"
        }
      },
      "entity": "4b5bac552b73526ebaf6010574c39b76",
      "parked": false
    }
  ]
}
```

Asset Fields (each item in data array)

Field	Type	Description
`ID`	string	Account entity ID
`accountActive`	boolean	Whether the account hosting the asset is active
`assetName`	string	Asset display name
`assetType`	string	Asset type
`data`	AssetData	Live data associated with the asset
`entity`	string	Asset entity ID
`parked`	boolean	Whether the asset is currently parked

AssetData Fields

Field	Type	Description
`location`	AssetLocation	Last valid location reading
`odometer`	AssetOdometer	Last valid odometer reading

AssetLocation Fields

Field	Type	Description
`alt`	number	Altitude (metres)
`heading`	number	Heading (degrees)
`lat`	number	Latitude (degrees)
`lng`	number	Longitude (degrees)
`received`	string	ISO 8601 time the reading was received
`speed`	number	Speed (km/h)
`timestamp`	string	ISO 8601 time the reading was taken

AssetOdometer Fields

Field	Type	Description
`odometer`	number	Odometer value (km)
`received`	string	ISO 8601 time the reading was received
`timestamp`	string	ISO 8601 time the reading was taken

Assets

GET /api/v1/assets/{assetID}

Get asset by ID

Retrieves a single asset by ID, scoped to the authenticated API key

Headers

Name	Required	Description
X-API-Key	Yes	API Key

Path Parameters

Name	Required	Description
assetID	Yes	Asset ID

Example Response (200)


```
{
  "data": {
    "ID": "27ea549ffb505fcfbac7b7208131a7c2",
    "accountActive": true,
    "assetName": "Example Asset",
    "assetType": "trailer",
    "data": {
      "location": {
        "alt": 12.3,
        "heading": 90,
        "lat": -33.8688,
        "lng": 151.2093,
        "received": "2026-09-29T00:36:53Z",
        "speed": 43.2,
        "timestamp": "2026-09-29T00:35:47Z"
      },
      "odometer": {
        "odometer": 27000,
        "received": "2026-09-29T00:36:53Z",
        "timestamp": "2026-09-29T00:35:47Z"
      }
    },
    "entity": "4b5bac552b73526ebaf6010574c39b76",
    "parked": false
  }
}
```

Asset Fields

Field	Type	Description
`ID`	string	Account entity ID
`accountActive`	boolean	Whether the account hosting the asset is active
`assetName`	string	Asset display name
`assetType`	string	Asset type
`data`	AssetData	Live data associated with the asset
`entity`	string	Asset entity ID
`parked`	boolean	Whether the asset is currently parked

AssetData Fields

Field	Type	Description
`location`	AssetLocation	Last valid location reading
`odometer`	AssetOdometer	Last valid odometer reading

AssetLocation Fields

Field	Type	Description
`alt`	number	Altitude (metres)
`heading`	number	Heading (degrees)
`lat`	number	Latitude (degrees)
`lng`	number	Longitude (degrees)
`received`	string	ISO 8601 time the reading was received
`speed`	number	Speed (km/h)
`timestamp`	string	ISO 8601 time the reading was taken

AssetOdometer Fields

Field	Type	Description
`odometer`	number	Odometer value (km)
`received`	string	ISO 8601 time the reading was received
`timestamp`	string	ISO 8601 time the reading was taken