

FleetPulse Developer Guide

- [Overview](#)
- [Terminology](#)
 - [Common Terms](#)
 - [Supported Sensor Types](#)
- [Technologies](#)
 - [GraphQL](#)
 - [Querying Flexibly With GraphQL](#)
 - [REST](#)
 - [Experimenting with the API](#)
 - [Consuming FleetPulse API From a Back-End API](#)
 - [Consuming FleetPulse API In a Web APP](#)
- [Security](#)
 - [Authentication](#)
 - [Example Authentication Request](#)
 - [Example Authentication Response](#)
- [Customer Master Data](#)
 - [Getting High-Level Details About Your Organization](#)
 - [Example Get Customers Request](#)
 - [Example Get Customers Response](#)
- [Vehicle Master Data](#)
 - [Listing Vehicles Registered With FleetPulse](#)
 - [Example List Vehicles Request](#)
 - [Example List Vehicles Response](#)
 - [Getting Details on a Specific Vehicle and its Current State](#)
 - [Get Vehicle Example Request](#)
 - [Get Vehicle Example Response](#)
- [Vehicle Transactional Data](#)
 - [Trailer History](#)
 - [Trailer History Example Request](#)
 - [Trailer History Example Response](#)
 - [Search For Trailers By Location](#)
 - [Search For Trailers By Location Example Request](#)
 - [Search For Trailers By Location Example Response](#)
 - [Fleet-Wide Trailer Health](#)
 - [Fleet-Wide Trailer Health Example Request](#)
 - [Fleet-Wide Trailer Health Example Response](#)
- [Geofencing](#)
 - [Listing Geofences](#)
 - [List Geofences Example Request](#)
 - [List Geofences Example Response](#)
 - [Creating Radius Geofences](#)
 - [Create Radius Geofence Example Request](#)
 - [Create Radius Geofence Example Response](#)
 - [Creating Polygon Geofences](#)
 - [Create Polygon Geofence Example Request](#)
 - [Create Polygon Geofence Example Response](#)
 - [Deleting Geofences](#)
 - [Delete Geofence Example Request](#)
- [Alerts](#)
 - [Active Alerts](#)
 - [List Active Alerts Example Request](#)

- [List Active Alerts Example Response](#)
- [Alert Details](#)
- [Alert Details Example Request](#)
- [Alert Details Example Response](#)
- [Triggered Alert Data](#)
- [Search Alert Data Example Request](#)
- [Search Alert Data Example Response](#)

Overview

Welcome to the FleetPulse developer guide. This document walks through the mechanics of common use cases of the FleetPulse API.

- An HTML version of this document can be found [here](#).
- A PDF version of this document can be found [here](#).
- The corresponding autogenerated GraphQL documentation for FleetPulse can be found [here](#) Always refer to that site for the latest GraphQL definitions.

Terminology

Common Terms

Term	Definition
Customer	Your organization, as registered with FleetPulse
Vehicle/Asset	A trailer installed with a FleetPulse-compatible sensor box and sensors
Sensor Box	A cellular-network connected IoT device which collects sensor data from one or more sensors throughout the vehicle and propogates to FleetPulse
Sensor	A physical sensor device can have one or more "logical" sensors. For example, an ABS system can generate both ABS fault readings and odometer readings. We treat each of these as its own sensor. Most physical sensor devices map to a single sensor. E.g. a door sensor.
Sensor Sample/Reading	All the details of an event or state of a sensor at a given point in time. E.g. a type of fault changing between active/inactive or the current bogie weight.

Supported Sensor Types

FleetPulse tracks data for a variety of sensor types installed on your fleet.

ID	Name	Description
7	ABS System Fault	Brake System Fault
36	Door Status	Door Open/Closed Status
5	Enhanced Weight	Moving Average of Net Cargo Weight--Adjusted for Fluctuations in Road Conditions (Grades, Bumps)
1	GPS	GPS Coordinates

ID	Name	Description
16	GPS Odometer	Cumulative Odometer as Estimated by GPS History
8	Light System Fault	Vehicle Electrical Light System Fault
6	Odometer	Cumulative ABS-Tracked Odometer (When Installed, More Accurate than GPS Odometer)
18	Sensor Box	Sensor Box-level events and metrics such as wake/sleep events, which battery is powering the sensor box, and trailer movement.
13	Sensor Box Battery	Sensor Boxes can switch between a dedicated battery and the tractor's battery. This sensor reads the sensor box's dedicated battery's amperage.
10	Tether Change	Tractor-trailer tether/un-tether state.
15	Tire Inflation Status	Tire leak status
14	Tractor Battery	Tractor battery amperage

Technologies

The FleetPulse API serves all data statelessly over HTTPS. The FleetPulse API URL is <https://my.fleetpulse.com>

GraphQL

Most data is served over [GraphQL](#) queries. Writes to the FleetPulse API for activities like user management, trailer management, setting alerts and geofences are done via GraphQL mutations. [The FleetPulse GraphQL API specification can be found here.](#)

Querying Flexibly With GraphQL

REST and GraphQL both return a schema of data for each different type of call (endpoint and query, respectively). However, the return schema for REST is "fixed" for each endpoint. For example, if a hypothetical REST endpoint for `/things` returns:

```
{
  "data": {
    "things": [
      {
        "id": "1",
        "name": "foo",
        "createdBy": "John Smith"
      },
      {
        "id": "2",
        "name": "bar",
        "createdBy": "Jane Doe"
      }
    ]
  }
}
```

but you as an API client only need the list of names, there's no way to convey over REST to omit the other fields without requiring an additional, less verbose endpoint.

An analogous hypothetical GraphQL query returning the same response would look like:

```
query {
  things {
    id
    name
    createdBy
  }
}
```

However, GraphQL has baked in support for you to convey to the API that you only need the things' names--which means fewer varieties of API calls for you to develop against. To request only the names, you would write:

```
query {
  things {
    name
  }
}
```

Which would send only the needed subset of data over the network:

```
{
  "data": {
    "things": [
      {
        "name": "foo"
      },
      {
        "name": "bar"
      }
    ]
  }
}
```

The above are not real APIs but serve to illustrate that you can add/omit supported fields from the GraphQL queries described in this document to fit your use cases. The queries throughout this document include commonly-used sets of fields.

Always refer to the latest, full FleetPulse GraphQL API specification [here](#).

REST

A small subset of the FleetPulse API is served over REST for activities like authentication and CSV downloads.

Experimenting with the API

Code examples in this document are showed in cURL for simplicity and cross-OS compatability but any HTTP IDE of your choice will work for experimenting with the API. [Insomnia](#), [Postman](#), [GraphiQL](#) are all great, free options.

Consuming FleetPulse API From a Back-End API

If you are writing an API that communicates directly with the FleetPulse API, any HTTP/REST library or framework will work for both GraphQL queries/mutations and REST endpoints.

Consuming FleetPulse API In a Web APP

If you are consuming the FleetPulse API in an existing or new web app, any Javascript HTTP/REST library or framework will work for both GraphQL queries/mutations and REST endpoints. However, for GraphQL features, you will likely achieve more

readable, shorter code and a better developer experience by leveraging any of the rich front-end GraphQL libraries/frameworks. Popular choices are: [Apollo Client](#), [Relay](#), [urql](#), and [graphql.js](#). If you are [React](#), most of these have a react-specific implementation to further decrease your development effort.

Security

Authentication

Users and API clients authenticate with FleetPulse using standard, basic, username/password auth which returns a [JWT](#). Future implementations will support standard [basic authentication](#). Subsequent calls to the FleetPulse API are authorized using that JWT in the `Authorization` header using [bearer token authorization](#). [jwt.io](#) is a fantastic resource for quickly and safely inspecting JWTs in the course of your development. Your JWT will only allow you to see data for your organization.

Example Authentication Request

```
curl --request GET \  
  --url https://my.fleetpulse.com/auth/v1/loginV2 \  
  --header 'authorization: Basic <base64 username:password>'
```

Example Authentication Response

```
{  
  "token": "<your jwt>"  
}
```

Customer Master Data

Getting High-Level Details About Your Organization

Example Get Customers Request

```
QUERY="$(echo 'query {  
  customers {  
    id  
    shortName  
    longName  
  }  
'})"  
  
curl --request POST \  
  --url https://my.fleetpulse.com/graphql \  
  -H 'authorization: <your jwt>' \  
  -H 'content-type: application/json' \  
  -d '{"query": "${QUERY}"}'
```

Example Get Customers Response

```
{  
  "data": {  
    "customers": [  
      {  
        "id": "1",  
        "shortName": "Acme",  
        "longName": "Acme Corporation"  
      }  
    ]  
  }  
}
```

```
    "longName": "Acme Corporation"
  }
]
}
}
```

Vehicle Master Data

Listing Vehicles Registered With FleetPulse

Example List Vehicles Request

```
QUERY="$(echo 'query {
  fleetStatus(input: {
    vehicleSensorBoxActive: true,
    recordedAfterVehicleSensorBoxPaired: true,
    customerVehicleActive: true
  }) {
    trailers {
      sensorBoxId
      extMfgrSerialNum
      sensorBoxMfgrId
      vehicleSensorBoxId
      vehicleId
      vin
      customerVehicleId
      unitId
      lastReportedAt
      gps {
        recorded
        lat
        lng
      }
      enhancedWeight {
        recorded
        weightLbs
        netWeightLbs
        isLoaded
        restingBogieWeightLbs
      }
      odometer {
        recorded
        miles
        adjustedMiles
      }
      abs {
        activeCount
        inactiveCount
        samples {
          recorded
          active
          faultTypeId
          sensorBoxMfgrFaultTypeId
          description
          blinkCode
          spn
          fmi
          suspectComponent
          faultDescription
          itpFunction
          cause
          repairInformation
        }
      }
      lights {
        activeCount
      }
    }
  }
}
```

```

        inactiveCount
        samples {
            recorded
            active
            faultTypeId
            sensorBoxMfgrFaultTypeId
            description
        }
    }
    tether {
        recorded
        tethered
    }
    door {
        recorded
        open
    }
    sensorBoxBattery{
        recorded
        millivolts
    }
    tractorBattery {
        recorded
        millivolts
    }
    tis {
        recorded
        statusId
        description
    }
    gpsOdometer {
        recorded
        miles
        adjustedMiles
    }
}
}
}
}')"
```

```

curl --request POST \
  --url https://my.fleetpulse.com/graphql \
  -H 'authorization: <your jwt>' \
  -H 'content-type: application/json' \
  -d "{ \"query\": \"${QUERY}\"}"
```

Example List Vehicles Response

```

{
  "data": {
    "fleetStatus": {
      "trailers": [
        {
          "sensorBoxId": "3a3cd46e-65ba-43b1-9d54-82e8af708d66",
          "extMfgrSerialNum": "1370860091",
          "sensorBoxMfgrId": 2,
          "vehicleSensorBoxId": "1b321c27-2eec-485a-8278-86bbcab2763c",
          "vehicleId": "f2d73ada-3600-4bc3-b96b-4a50a90c59a4",
          "vin": "1GR1P0620LJ156066",
          "customerVehicleId": "bb0c7fa7-2fa6-4f58-ae06-07ed3eed18c9",
          "unitId": "R&D_1",
          "lastReportedAt": "2022-03-18T14:55:07Z",
          "gps": {
            "recorded": "2022-03-17T14:55:34Z",
            "lat": 32.1649733,
            "lng": -81.212386
          },
          "enhancedWeight": {
```

```

        "recorded": "2022-03-07T11:59:03Z",
        "weightLbs": 7751,
        "netWeightLbs": null,
        "isLoaded": false,
        "restingBogieWeightLbs": 9260
    },
    "odometer": {
        "recorded": "2022-03-07T11:59:03Z",
        "miles": 16312.24562905,
        "adjustedMiles": 16312.24562905
    },
    "abs": null,
    "lights": null,
    "tether": {
        "recorded": "2022-03-18T14:55:07Z",
        "tethered": false
    },
    "door": {
        "recorded": "2022-03-17T12:10:29Z",
        "open": true
    },
    "sensorBoxBattery": {
        "recorded": "2022-03-18T14:55:07Z",
        "millivolts": 12683
    },
    "tractorBattery": {
        "recorded": "2022-03-07T11:59:03Z",
        "millivolts": null
    },
    "tis": {
        "recorded": "2022-03-07T11:59:03Z",
        "statusId": 2,
        "description": "stable"
    },
    "gpsOdometer": {
        "recorded": "2022-03-17T12:10:29Z",
        "miles": 15401.152546000001,
        "adjustedMiles": 15401.152546000001
    }
}
    ]
}
}
}

```

Getting Details on a Specific Vehicle and its Current State

Get Vehicle Example Request

```

QUERY=$(echo 'query {
  vehicleDetail(vin: \"<some vin>\") {
    id
    vin
    vehicleType {
      id
      shortName
      longName
    }
    vehicleSensorBoxes {
      id
    }
    specs
    vehicleSpecs {
      product
      mfgPlant
      year
      packagedOptions
    }
  }
}' | tr -d '\n')

```

```
    serialPlate
    overallHeight
    overallLength
    overallWidth
    upperCouplerHeight
    kingpinLocation
    landingGearLocation
    wedge
    undercarriageDesign
}
ageInDays
orderNumber
lastCargoImage {
  id
  trailerId
  level
  imageUrl
  rawResponse
  capturedAt
  created
}
loadStatus {
  status
  cargoWeight
}
sensorDiagnostics {
  gpsDiagnostics {
    lastCommunicationAt
    displayLastCommunicationAt
    statusIcon
    vehicleDiagnosticTypeEnum
    vehicleDiagnosticType {
      title
      icon
      key
    }
  }
  sensorManufacturer
  satelliteCount
  vehicleAddress {
    displayAddressLine1
    displayCityStateAndZip
    location {
      displayLatAndLong
      displayRecordedAt
      lat
      lng
    }
  }
}
}
tetherStatusDiagnostics {
  lastCommunicationAt
  displayLastCommunicationAt
  statusIcon
  vehicleDiagnosticTypeEnum
  vehicleDiagnosticType {
    title
    icon
    key
  }
  isTethered
}
bogieWeightDiagnostics {
  lastCommunicationAt
  displayLastCommunicationAt
  statusIcon
  vehicleDiagnosticTypeEnum
  vehicleDiagnosticType {
    title
    icon
    key
  }
}
```

```

        weightLbs
        isLoaded
    }
    tireInflationStatusDiagnostic {
        lastCommunicationAt
        displayLastCommunicationAt
        statusIcon
        vehicleDiagnosticTypeEnum
        vehicleDiagnosticType {
            title
            icon
            key
        }
        manufacturer
        status
    }
    doorStatusDiagnostics {
        lastCommunicationAt
        displayLastCommunicationAt
        statusIcon
        vehicleDiagnosticTypeEnum
        vehicleDiagnosticType {
            title
            icon
            key
        }
        manufacturer
        status
    }
    fleetPulseDeviceDiagnostics {
        lastCommunicationAt
        displayLastCommunicationAt
        statusIcon
        vehicleDiagnosticTypeEnum
        vehicleDiagnosticType {
            title
            icon
            key
        }
        deviceManufacturer
        batteryVoltage
        gpsOdometerMiles
    }
    absDiagnostics {
        lastCommunicationAt
        displayLastCommunicationAt
        statusIcon
        vehicleDiagnosticTypeEnum
        vehicleDiagnosticType {
            title
            icon
            key
        }
    }
    faultCount
    activeFaultCount
    activeFaults {
        lastCommunicationAt
        displayLastCommunicationAt
        statusIcon
        vehicleDiagnosticTypeEnum
        sensorSampleId
        sensorTypeId
        sensorTypeName
        isFaultActive
        sensorBoxMfgrFaultType {
            id
            sensorBoxMfgr {
                id
                shortName
                longName
            }
        }
    }

```

```

        faultType {
            id
            description
            cause
            repairInformation
            spn
            fmi
            blinkCode
            suspectComponent
        }
        extFaultTypeId
        extDescription
    }
    sensorManufacturer
    firmwareHardware
    configuration
}
inactiveFaults {
    lastCommunicationAt
    displayLastCommunicationAt
    statusIcon
    vehicleDiagnosticTypeEnum
    sensorSampleId
    sensorTypeId
    sensorTypeName
    isFaultActive
    sensorBoxMfgrFaultType {
        id
        sensorBoxMfgr {
            id
            shortName
            longName
        }
        faultType {
            id
            description
            cause
            repairInformation
            spn
            fmi
            blinkCode
            suspectComponent
        }
        extFaultTypeId
        extDescription
    }
    sensorManufacturer
    firmwareHardware
    configuration
}
absAmberWarningSignalLamp {
    sensorSampleId
    isOn
}
odometerDiagnostics {
    miles
    displayMiles
}
}
lightsDiagnostics {
    lastCommunicationAt
    displayLastCommunicationAt
    statusIcon
    vehicleDiagnosticTypeEnum
    vehicleDiagnosticType {
        title
        icon
        key
    }
}
faultCount
activeFaultCount

```

```
activeFaults {
  lastCommunicationAt
  displayLastCommunicationAt
  statusIcon
  vehicleDiagnosticTypeEnum
  sensorSampleId
  sensorTypeId
  sensorTypeName
  isFaultActive
  sensorBoxMfgrFaultType {
    id
    sensorBoxMfgr {
      id
      shortName
      longName
    }
    faultType {
      id
      description
      cause
      repairInformation
      spn
      fmi
      blinkCode
      suspectComponent
    }
    extFaultTypeId
    extDescription
  }
  sensorManufacturer
  firmwareHardware
  configuration
}
inactiveFaults {
  lastCommunicationAt
  displayLastCommunicationAt
  statusIcon
  vehicleDiagnosticTypeEnum
  sensorSampleId
  sensorTypeId
  sensorTypeName
  isFaultActive
  sensorBoxMfgrFaultType {
    id
    sensorBoxMfgr {
      id
      shortName
      longName
    }
    faultType {
      id
      description
      cause
      repairInformation
      spn
      fmi
      blinkCode
      suspectComponent
    }
    extFaultTypeId
    extDescription
  }
  sensorManufacturer
  firmwareHardware
  configuration
}
groteCommFaults {
  lastCommunicationAt
  displayLastCommunicationAt
  statusIcon
  vehicleDiagnosticTypeEnum
```

```
sensorSampleId
sensorTypeId
sensorTypeName
isFaultActive
sensorBoxMfgrFaultType {
  id
  sensorBoxMfgr {
    id
    shortName
    longName
  }
  faultType {
    id
    description
    cause
    repairInformation
    spn
    fmi
    blinkCode
    suspectComponent
  }
  extFaultTypeId
  extDescription
}
sensorManufacturer
firmwareHardware
configuration
}
activeBlackCircuitFaults {
  lastCommunicationAt
  displayLastCommunicationAt
  statusIcon
  vehicleDiagnosticTypeEnum
  sensorSampleId
  sensorTypeId
  sensorTypeName
  isFaultActive
  sensorBoxMfgrFaultType {
    id
    sensorBoxMfgr {
      id
      shortName
      longName
    }
    faultType {
      id
      description
      cause
      repairInformation
      spn
      fmi
      blinkCode
      suspectComponent
    }
    extFaultTypeId
    extDescription
  }
  sensorManufacturer
  firmwareHardware
  configuration
}
activeYellowCircuitFaults {
  lastCommunicationAt
  displayLastCommunicationAt
  statusIcon
  vehicleDiagnosticTypeEnum
  sensorSampleId
  sensorTypeId
  sensorTypeName
  isFaultActive
  sensorBoxMfgrFaultType {
```

```

        id
        sensorBoxMfgr {
            id
            shortName
            longName
        }
        faultType {
            id
            description
            cause
            repairInformation
            spn
            fmi
            blinkCode
            suspectComponent
        }
        extFaultTypeId
        extDescription
    }
    sensorManufacturer
    firmwareHardware
    configuration
}
activeRedCircuitFaults {
    lastCommunicationAt
    displayLastCommunicationAt
    statusIcon
    vehicleDiagnosticTypeEnum
    sensorSampleId
    sensorTypeId
    sensorTypeName
    isFaultActive
    sensorBoxMfgrFaultType {
        id
        sensorBoxMfgr {
            id
            shortName
            longName
        }
        faultType {
            id
            description
            cause
            repairInformation
            spn
            fmi
            blinkCode
            suspectComponent
        }
        extFaultTypeId
        extDescription
    }
    sensorManufacturer
    firmwareHardware
    configuration
}
activeGreenCircuitFaults {
    lastCommunicationAt
    displayLastCommunicationAt
    statusIcon
    vehicleDiagnosticTypeEnum
    sensorSampleId
    sensorTypeId
    sensorTypeName
    isFaultActive
    sensorBoxMfgrFaultType {
        id
        sensorBoxMfgr {
            id
            shortName
            longName
        }
        faultType {
            id
            description
            cause
            repairInformation
            spn
            fmi
            blinkCode
            suspectComponent
        }
        extFaultTypeId
        extDescription
    }
    sensorManufacturer
    firmwareHardware
    configuration
}

```


Vehicle Transactional Data

Trailer History

Trailer History Example Request

```
QUERY="$(echo 'query {
  {vehicleTrackingV2
    (vin: "1GR1P0620LJ156066"
      , pageSize: 100
      , pageNumber: 0
    ) {
      activityLog {
        vin
        recorded
        latitude
        longitude
        reverseGeoFullAddress
        tetherStatus
        absFaultDescription
        amberLampStatus
        tisStatus
        loaded
        loadWeight
        batteryVoltageTrailer
        doorOpened
        lightRed
        lightGreen
        lightYellow
        lightBlack
        lightBrown
        odometerAbs
        odometerGps
      }
    }
  }
}')"
```

```
curl --request POST \
  --url https://my.fleetpulse.com/graphql \
  -H 'authorization: <your jwt>' \
  -H 'content-type: application/json' \
  -d '{"query": "'.$QUERY'"}
```

Trailer History Example Response

```
{
  "data": {
    "vehicleTrackingV2": {
      "activityLog": [
        {
          "vin": "1GR1P0620LJ156066",
          "recorded": "2021-06-15T16:34:09Z",
          "latitude": null,
          "longitude": null,
          "reverseGeoFullAddress": null,
          "tetherStatus": "Untethered",
          "absFaultDescription": null,
          "amberLampStatus": null,
          "tisStatus": null,
          "loaded": null,
          "loadWeight": null,
          "batteryVoltageTrailer": 38.0,
          "doorOpened": null,
          "lightRed": null,
          "lightGreen": null,
```

```
    "lightYellow": null,
    "lightBlack": null,
    "lightBrown": null,
    "odometerAbs": null,
    "odometerGps": null
  },
  {
    "vin": "1GR1P0620LJ156066",
    "recorded": "2021-06-15T16:32:04Z",
    "latitude": null,
    "longitude": null,
    "reverseGeoFullAddress": null,
    "tetherStatus": "Untethered",
    "absFaultDescription": null,
    "amberLampStatus": null,
    "tisStatus": null,
    "loaded": null,
    "loadWeight": null,
    "batteryVoltageTrailer": null,
    "doorOpened": null,
    "lightRed": null,
    "lightGreen": null,
    "lightYellow": null,
    "lightBlack": null,
    "lightBrown": null,
    "odometerAbs": null,
    "odometerGps": null
  },
  {
    "vin": "1GR1P0620LJ156066",
    "recorded": "2021-06-15T16:30:03Z",
    "latitude": null,
    "longitude": null,
    "reverseGeoFullAddress": null,
    "tetherStatus": "Tethered",
    "absFaultDescription": null,
    "amberLampStatus": null,
    "tisStatus": null,
    "loaded": null,
    "loadWeight": null,
    "batteryVoltageTrailer": null,
    "doorOpened": null,
    "lightRed": null,
    "lightGreen": null,
    "lightYellow": null,
    "lightBlack": null,
    "lightBrown": null,
    "odometerAbs": null,
    "odometerGps": 200.888921
  },
  {
    "vin": "1GR1P0620LJ156066",
    "recorded": "2021-06-15T16:28:48Z",
    "latitude": null,
    "longitude": null,
    "reverseGeoFullAddress": null,
    "tetherStatus": "Untethered",
    "absFaultDescription": null,
    "amberLampStatus": null,
    "tisStatus": null,
    "loaded": null,
    "loadWeight": null,
    "batteryVoltageTrailer": null,
    "doorOpened": null,
    "lightRed": null,
    "lightGreen": null,
    "lightYellow": null,
    "lightBlack": null,
    "lightBrown": null,
    "odometerAbs": null,
    "odometerGps": null
  }
}
```

```
},
{
  "vin": "1GR1P0620LJ156066",
  "recorded": "2021-06-15T15:27:21Z",
  "latitude": null,
  "longitude": null,
  "reverseGeoFullAddress": null,
  "tetherStatus": "Untethered",
  "absFaultDescription": null,
  "amberLampStatus": null,
  "tisStatus": null,
  "loaded": null,
  "loadWeight": null,
  "batteryVoltageTrailer": 48.0,
  "doorOpened": null,
  "lightRed": null,
  "lightGreen": null,
  "lightYellow": null,
  "lightBlack": null,
  "lightBrown": null,
  "odometerAbs": null,
  "odometerGps": null
},
{
  "vin": "1GR1P0620LJ156066",
  "recorded": "2021-06-15T15:25:21Z",
  "latitude": null,
  "longitude": null,
  "reverseGeoFullAddress": null,
  "tetherStatus": "Tethered",
  "absFaultDescription": null,
  "amberLampStatus": null,
  "tisStatus": null,
  "loaded": null,
  "loadWeight": null,
  "batteryVoltageTrailer": null,
  "doorOpened": null,
  "lightRed": null,
  "lightGreen": null,
  "lightYellow": null,
  "lightBlack": null,
  "lightBrown": null,
  "odometerAbs": null,
  "odometerGps": 200.888921
},
{
  "vin": "1GR1P0620LJ156066",
  "recorded": "2021-06-15T15:24:44Z",
  "latitude": null,
  "longitude": null,
  "reverseGeoFullAddress": null,
  "tetherStatus": "Untethered",
  "absFaultDescription": null,
  "amberLampStatus": null,
  "tisStatus": null,
  "loaded": null,
  "loadWeight": null,
  "batteryVoltageTrailer": null,
  "doorOpened": null,
  "lightRed": null,
  "lightGreen": null,
  "lightYellow": null,
  "lightBlack": null,
  "lightBrown": null,
  "odometerAbs": null,
  "odometerGps": null
},
{
  "vin": "1GR1P0620LJ156066",
  "recorded": "2021-06-15T15:22:40Z",
  "latitude": null,
```

```

        "longitude": null,
        "reverseGeoFullAddress": null,
        "tetherStatus": "Untethered",
        "absFaultDescription": null,
        "amberLampStatus": null,
        "tisStatus": null,
        "loaded": null,
        "loadWeight": null,
        "batteryVoltageTrailer": 38.0,
        "doorOpened": null,
        "lightRed": null,
        "lightGreen": null,
        "lightYellow": null,
        "lightBlack": null,
        "lightBrown": null,
        "odometerAbs": null,
        "odometerGps": null
    },
    {
        "vin": "1GR1P0620LJ156066",
        "recorded": "2021-06-15T15:20:39Z",
        "latitude": null,
        "longitude": null,
        "reverseGeoFullAddress": null,
        "tetherStatus": "Tethered",
        "absFaultDescription": null,
        "amberLampStatus": null,
        "tisStatus": null,
        "loaded": null,
        "loadWeight": null,
        "batteryVoltageTrailer": null,
        "doorOpened": null,
        "lightRed": null,
        "lightGreen": null,
        "lightYellow": null,
        "lightBlack": null,
        "lightBrown": null,
        "odometerAbs": null,
        "odometerGps": 200.888921
    },
    {
        "vin": "1GR1P0620LJ156066",
        "recorded": "2021-06-15T15:18:22Z",
        "latitude": 32.1650975,
        "longitude": -81.2124305,
        "reverseGeoFullAddress": "131 Technology Circle, Savannah, Georgia 31408, United States",
        "tetherStatus": "Untethered",
        "absFaultDescription": null,
        "amberLampStatus": null,
        "tisStatus": null,
        "loaded": null,
        "loadWeight": null,
        "batteryVoltageTrailer": null,
        "doorOpened": null,
        "lightRed": null,
        "lightGreen": null,
        "lightYellow": null,
        "lightBlack": null,
        "lightBrown": null,
        "odometerAbs": null,
        "odometerGps": null
    }
}
}
}

```

Search For Trailers By Location

Search For Trailers By Location Example Request

```

QUERY="$(echo 'query {
  findVehicles(input: {lat: <some latitude>, lng: <some longitude>}) {
    all {
      total
      data {
        vehicle {
          id
          vin
          location {
            lat
            lng
            displayLatAndLong
            displayRecordedAt
          }
        }
      }
    }
  }
  nearby {
    data {
      vehicle {
        id
        vin
        location {
          lat
          lng
          displayLatAndLong
          displayRecordedAt
        }
      }
    }
  }
  recent {
    data {
      vehicle {
        id
        vin
        location {
          lat
          lng
          displayLatAndLong
          displayRecordedAt
        }
      }
    }
  }
}'")"

```

```

curl --request POST \
  --url https://my.fleetpulse.com/graphql \
  -H 'authorization: <your jwt>' \
  -H 'content-type: application/json' \
  -d '{"query": \"$QUERY\"}'

```

Search For Trailers By Location Example Response

```

{
  "data": {
    "findVehicles": {
      "all": {
        "total": 2,
        "data": [
          {
            "vehicle": {
              "id": "13cee1ac-13e2-464c-8a31-cdb258fc86bc",
              "vin": "SCAZS42A1GCX14122",
              "location": {

```



```

QUERY="$(echo 'query {
  vehicleFleetStatistics {
    healthyVehiclesCount
    unhealthyVehiclesCount
    sensorTypeStatistics {
      vehiclesCount
      sensorType {
        id
        longName
      }
    }
  }
}'")"

curl --request POST \
  --url https://my.fleetpulse.com/graphql \
  -H 'authorization: <your jwt>' \
  -H 'content-type: application/json' \
  -d "{ \"query\": \"$QUERY\"}"

```

Fleet-Wide Trailer Health Example Response

```

{
  "data": {
    "vehicleFleetStatistics": {
      "healthyVehiclesCount": 35,
      "unhealthyVehiclesCount": 26,
      "sensorTypeStatistics": [
        {
          "vehiclesCount": 26,
          "sensorType": {
            "id": "8",
            "longName": "Light System Fault"
          }
        },
        {
          "vehiclesCount": 2,
          "sensorType": {
            "id": "7",
            "longName": "ABS System Fault"
          }
        }
      ]
    }
  }
}

```

Geofencing

FleetPulse supports managing radius (circular) and arbitrary polygon geofences for your fleet. Geofence event notifications are delivered by email.

Listing Geofences

List Geofences Example Request

```

QUERY="$(echo 'query {
  geofencesV2 {
    geofenceId
    radiusMeters
    geofenceTypeId
  }
}'")"

```

```

    geofenceDeactivated
    customerGeofenceId
    customerId
    name
    description
    customerGeofenceDeactivated
    boundaryCoordinates {
      latitude
      longitude
    }
    centerPointCoordinate {
      latitude
      longitude
    }
  }
}
}')"
```

```

curl --request POST \
  --url https://my.fleetpulse.com/graphql \
  -H 'authorization: <your jwt>' \
  -H 'content-type: application/json' \
  -d '{"query": "$QUERY"}'
```

List Geofences Example Response

```

{
  "data": {
    "geofencesV2": [
      {
        "geofenceId": "740a8faf-1f69-4c65-9417-954740e651e3",
        "radiusMeters": 20000.0,
        "geofenceTypeId": 1,
        "geofenceDeactivated": null,
        "customerGeofenceId": "cb421e5e-de0d-4a9d-a492-91e616df7cef",
        "customerId": 0,
        "name": "Houston 200km Radius",
        "description": "Houston 200km Radius",
        "customerGeofenceDeactivated": null,
        "boundaryCoordinates": null,
        "centerPointCoordinate": {
          "latitude": 29.7604,
          "longitude": -95.3698
        }
      },
      {
        "geofenceId": "3365edfb-0fe5-43e6-b4a9-290d3c98c9dc",
        "radiusMeters": null,
        "geofenceTypeId": 2,
        "geofenceDeactivated": null,
        "customerGeofenceId": "2293b5a7-d5e8-4d6b-98d0-5a3e34523331",
        "customerId": 0,
        "name": "test",
        "description": "",
        "customerGeofenceDeactivated": null,
        "boundaryCoordinates": [
          {
            "latitude": 42.7843150113756,
            "longitude": -108.72070275
          },
          {
            "latitude": 44.375895128647045,
            "longitude": -108.193359
          },
          {
            "latitude": 44.751609810531804,
            "longitude": -106.2158199375
          }
        ]
      }
    ]
  }
}
```

```

        "latitude": 43.041794498605576,
        "longitude": -104.3701168125
      },
      {
        "latitude": 42.428525033683655,
        "longitude": -106.874999625
      },
      {
        "latitude": 42.7843150113756,
        "longitude": -108.72070275
      }
    ],
    "centerPointCoordinate": null
  }
]
}
}

```

Creating Radius Geofences

Create Radius Geofence Example Request

```

QUERY="$(echo 'mutation {
  createGeofence(
    geofenceInput: {
      name: "Radius Geo",
      active: true,
      geofenceTypeId: 1
      radiusMeters: 100000,
      centerPointCoordinate: {
        latitude: 45.02930,
        longitude: -110.70703,
      }
    }
  }) {
    geofenceId
  }
}')"

curl --request POST \
  --url https://my.fleetpulse.com/graphql \
  -H 'authorization: <your jwt>' \
  -H 'content-type: application/json' \
  -d '{"query": \"$QUERY\"}'

```

Create Radius Geofence Example Response

```

{
  "data": {
    "createGeofence": {
      "geofenceId": "dc1cab3-1e7c-45d9-bf17-7dfa915b8d39"
    }
  }
}

```

Creating Polygon Geofences

Polygon geofences consist of 3 or more arbitrary coordinates which form a boundary.

Create Polygon Geofence Example Request

```

QUERY="$(echo 'mutation {
  geofenceInput: {
    name: "Polygon Geo",
    geofenceTypeId: 2,
    active: true,
    boundaryCoordinates: [
      {
        latitude: 43.64899454100273,
        longitude: -96.08642540625
      },
      {
        latitude: 42.428525033683655,
        longitude: -95.20751915625
      },
      {
        latitude: 43.07390378795281,
        longitude: -93.22998009375
      }
    ]
  }
}) {
  geofenceId
  name
  geofenceTypeId
  boundaryCoordinates {
    longitude
    latitude
  }
}'")

curl --request POST \
  --url https://my.fleetpulse.com/graphql \
  -H 'authorization: <your jwt>' \
  -H 'content-type: application/json' \
  -d "{ \"query\": \"$QUERY\"}"

```

Create Polygon Geofence Example Response

```

{
  "data": {
    "createGeofence": {
      "geofenceId": "cb4ec49c-f41a-4458-8b64-f24fde322417",
      "name": "Polygon Test 4",
      "geofenceTypeId": 2,
      "boundaryCoordinates": [
        {
          "longitude": -96.08642540625,
          "latitude": 43.6489945410027
        },
        {
          "longitude": -95.20751915625,
          "latitude": 42.428525033683655
        },
        {
          "longitude": -93.22998009375,
          "latitude": 43.07390378795281
        },
        {
          "longitude": -96.08642540625,
          "latitude": 43.64899454100273
        }
      ]
    }
  }
}

```

Deleting Geofences

If you have geofences you no longer need, they can be deleted with deleteGeofence.

Delete Geofence Example Request

```
QUERY="$(echo 'mutation {
  deleteGeofence(id: "39d82233-75bb-40f2-b4fa-fc98853bdc7f") {
    deletedCount
  }
}')"

curl --request POST \
  --url https://my.fleetpulse.com/graphql \
  -H 'authorization: <your jwt>' \
  -H 'content-type: application/json' \
  -d '{"query": "$QUERY"}'
```

Alerts

FleetPulse supports the viewing of alerts and alerts data via API.

Active Alerts

getAlerts (Get list of alerts on user account)

List Active Alerts Example Request

```
QUERY="$(echo 'query {
  getAlerts {
    id
    name
    customerId
    allVehicles
    vehicleIds
    vehicleGroupIds
    alertNotificationRateLimitMs
    oneTimeOnlyPerVehicle
    includeSensorData
    createdById
    notifyEmails
    created
    updated
    deactivated
  }
}')"

curl --request POST \
  --url https://my.fleetpulse.com/graphql \
  -H 'authorization: <your jwt>' \
  -H 'content-type: application/json' \
  -d '{"query": "$QUERY"}'
```

List Active Alerts Example Response

```
{
  "data": {
    "getAlerts": [
      {
        "id": "1ebc36cf-e050-4182-ba51-cb57f87bf9d6",
        "name": "battery alert",
```

```
"customerId": 9,
"allVehicles": true,
"vehicleIds": null,
"vehicleGroupIds": null,
>alertNotificationRateLimitMs": 86400000,
"oneTimeOnlyPerVehicle": false,
"includeSensorData": false,
"createdById": "c0a27cb8-d465-4e1c-aa10-8cd0be05299e",
"notifyEmails": [
  "test@test.com"
],
"created": "2024-08-02T15:48:08.739106Z",
"updated": "2024-08-02T15:48:08.738182Z",
"deactivated": null
},
{
  "id": "39b298b9-12d9-4d12-97db-2bf35e85e2b9",
  "name": "Load test",
  "customerId": 9,
  "allVehicles": true,
  "vehicleIds": null,
  "vehicleGroupIds": null,
 >alertNotificationRateLimitMs": 86400000,
  "oneTimeOnlyPerVehicle": false,
  "includeSensorData": false,
  "createdById": "c0a27cb8-d465-4e1c-aa10-8cd0be05299e",
  "notifyEmails": [
    "test@test.com"
  ],
  "created": "2024-08-02T15:47:54.657886Z",
  "updated": "2024-08-02T15:47:54.654569Z",
  "deactivated": null
},
{
  "id": "135c181c-948c-444e-bf09-1e1d749d3590",
  "name": "Lights Tests",
  "customerId": 9,
  "allVehicles": false,
  "vehicleIds": null,
  "vehicleGroupIds": [
    "4d33a4fe-bc19-463c-a5ac-931fbd050a37"
  ],
 >alertNotificationRateLimitMs": 86400000,
  "oneTimeOnlyPerVehicle": false,
  "includeSensorData": false,
  "createdById": "c0a27cb8-d465-4e1c-aa10-8cd0be05299e",
  "notifyEmails": [
    "test@test.com"
  ],
  "created": "2024-07-29T21:01:32.178502Z",
  "updated": "2024-07-29T21:01:32.177533Z",
  "deactivated": null
},
{
  "id": "94bc390c-abe2-428a-aeb7-262b5325efa7",
  "name": "ABS Fault Test",
  "customerId": 9,
  "allVehicles": true,
  "vehicleIds": null,
  "vehicleGroupIds": null,
 >alertNotificationRateLimitMs": 86400000,
  "oneTimeOnlyPerVehicle": false,
  "includeSensorData": false,
  "createdById": "c0a27cb8-d465-4e1c-aa10-8cd0be05299e",
  "notifyEmails": [
    "test@test.com"
  ],
  "created": "2024-07-29T21:01:04.244826Z",
  "updated": "2024-07-29T21:01:04.240988Z",
  "deactivated": null
}
}
```

```
}
  }
}
```

Alert Details

Alert Details Example Request

```
QUERY="$(echo 'query {
  getAlert(id: "1ebc36cf-e050-4182-ba51-cb57f87bf9d6") {
    id
    name
    customerId
    allVehicles
    vehicleIds
    vehicleGroupIds
    alertNotificationRateLimitMs
    oneTimeOnlyPerVehicle
    includeSensorData
    createdById
    notifyEmails
    created
    updated
    deactivated
  }
}')"
```

```
curl --request POST \
  --url https://my.fleetpulse.com/graphql \
  -H 'authorization: <your jwt>' \
  -H 'content-type: application/json' \
  -d "{ \"query\": \"$QUERY\"}"
```

Alert Details Example Response

```
{
  "data": {
    "getAlert": {
      "id": "1ebc36cf-e050-4182-ba51-cb57f87bf9d6",
      "name": "battery alert",
      "customerId": 9,
      "allVehicles": true,
      "vehicleIds": null,
      "vehicleGroupIds": null,
      "alertNotificationRateLimitMs": 86400000,
      "oneTimeOnlyPerVehicle": false,
      "includeSensorData": false,
      "createdById": "c0a27cb8-d465-4e1c-aa10-8cd0be05299e",
      "notifyEmails": [
        "test@test.com"
      ],
      "created": "2024-08-02T15:48:08.739106Z",
      "updated": "2024-08-02T15:48:08.738182Z",
      "deactivated": null
    }
  }
}
```

Triggered Alert Data

Note: The naming here is a little confusing. searchAlerts is for finding alerts that have been triggered and the data that triggered them. getAlerts and getAlert are for viewing what alerts are configured.

searchAlerts (Get the contents of any Alerts that were activated)

Search Alert Data Example Request

```
QUERY="$(echo 'query {
  searchAlerts(
    input: {
      pageSize: 10
      pageNumber: 1
      startDateTime: "2024-07-05T00:00:00+00:00"
      endDateTime: "2024-07-24T00:00:00+00:00"
      notifiedEmail: "test@test.com"
      alertNameSubstring: "My Test Alert"
    }
  ) {
    totalPages
    totalItems
    items {
      id
      rawMessageId
      recorded
      sensorBoxMfgrId
      sensorBoxId
      extMfgrSerialNum
      vehicleId
      vin
      customerId
      customerName
      reportTypeName
      latitude
      longitude
      altitude
      satellitesCount
      speed
      heading
      accelerometerMotion
      lightRed
      lightGreen
      lightYellow
      lightBlack
      lightBrown
      tisStatus
      tetherStatus
      odometerAbs
      odometerGps
      batteryVoltageTractor
      batteryVoltageTrailer
      doorOpened
      bogieWeight
      loaded
      loadWeight
      created
      unitId
      absMilesTraveled
      gpsMilesTraveled
      lightFaults
      absFaults
      brakeLiningInsufficientStatus
      amberLampStatus
      vehicleElectricalSupplyInsufficientStatus
      sensorBoxMfgrName
      reportTypeId
      withinCustomerGeofenceNames
      reverseGeoFullAddress
      absFaultLevelId
    }
  }
})'"
```

```

        absFaultDescription
        timeElapsedMs
        updated
        durationInStateMs
        idleEventCount
        idleTimeDurationMs
        adjOdometerGps
        adjOdometerAbs
        occupiedCustomerSpecificGeofenceIds
        occupiedGeofenceIds
        occupiedGeofences
        occupiedCustomerSpecificGeofences
        faultLevel
        tseFaults
        backupBatteryVoltage
        originalRecorded
        isRecordedCorrected
        tseFaultDetails
        wheelEndDetails
        tirePressureDetails
        reverseGeo
        absSeverityLevel
        thermoKingDetails
        ultrasonicCargoSensorVoltage
        cargoCameraImageS3Key
        cargoCameraFullness
        thermoKingAlarmCodes
        cargoCameraLat
        cargoCameraLong
        cargoCameraReverseGeo
        reeferFaultDetails
        reeferZoneTemperatureDetails
        reeferLatitude
        reeferLongitude
        reeferReverseGeo
        reeferStatusDetails
        reeferBatteryDetails
        reeferFuelLevelDetails
        reeferHoursDetails
        reeferTemperatureSensorsDetails
    }
}
}
}')"
```

```

curl --request POST \
  --url https://my.fleetpulse.com/graphql \
  -H 'authorization: <your jwt>' \
  -H 'content-type: application/json' \
  -d '{"query": "$QUERY"}'
```

Search Alert Data Example Response

```

{
  "data": {
    "searchAlerts": {
      "totalPages": 1,
      "totalItems": 2,
      "items": [
        {
          "id": "0f0422d3-9dd2-4944-a60d-c9883b944bc7",
          "rawMessageId": "e537883e-fb6e-497c-9cc7-ff2917b85e1a",
          "recorded": "2024-07-15T16:24:44Z",
          "sensorBoxMfgId": 6,
          "sensorBoxId": "b64cf17f-e396-48a5-9477-9be75d4fa5c9",
          "extMfgSerialNum": "1100240400007",
          "vehicleId": "af102ab3-b3a5-4750-a38b-67eb1840942e",
          "vin": "FPXT5300TEST00007",
          "customerId": 9,
        }
      ]
    }
  }
}
```

```
"customerName": "Morey Test Trailers",
"reportTypeName": "XT5300_DATA_EVENT_CODE",
"latitude": 32.016618,
"longitude": -81.10866,
"altitude": null,
"satellitesCount": null,
"speed": null,
"heading": null,
"accelerometerMotion": null,
"lightRed": null,
"lightGreen": null,
"lightYellow": null,
"lightBlack": null,
"lightBrown": null,
"tisStatus": null,
"tetherStatus": null,
"odometerAbs": null,
"odometerGps": null,
"batteryVoltageTractor": null,
"batteryVoltageTrailer": null,
"doorOpened": null,
"bogieWeight": null,
"loaded": null,
"loadWeight": null,
"created": "2024-07-15T16:24:55.06954Z",
"unitId": null,
"absMilesTraveled": null,
"gpsMilesTraveled": null,
"lightFaults": null,
"absFaults": null,
"brakeLiningInsufficientStatus": null,
"amberLampStatus": null,
"vehicleElectricalSupplyInsufficientStatus": null,
"sensorBoxMfrName": null,
"reportTypeId": 71,
"withinCustomerGeofenceNames": null,
"reverseGeoFullAddress": "4 Jackson Court, Savannah, GA 31405, United States of America",
"absFaultLevelId": null,
"absFaultDescription": null,
"timeElapsedMs": null,
"updated": "2024-07-15T16:24:55.06954Z",
"durationInStateMs": null,
"idleEventCount": null,
"idleTimeDurationMs": null,
"adjOdometerGps": null,
"adjOdometerAbs": null,
"occupiedCustomerSpecificGeofenceIds": null,
"occupiedGeofenceIds": null,
"occupiedGeofences": "{}",
"occupiedCustomerSpecificGeofences": "{}",
"faultLevel": null,
"tseFaults": null,
"backupBatteryVoltage": null,
"originalRecorded": null,
"isRecordedCorrected": null,
"tseFaultDetails": null,
"wheelEndDetails": null,
"tirePressureDetails": null,
"reverseGeo": "{\"place\":\"Savannah\",\"region\":\"Georgia\",\"address\":\"4 Jackson Cou",
"absSeverityLevel": null,
"thermoKingDetails": null,
"ultrasonicCargoSensorVoltage": null,
"cargoCameraImageS3Key": null,
"cargoCameraFullness": "Failed",
"thermoKingAlarmCodes": null,
"cargoCameraLat": null,
"cargoCameraLong": null,
"cargoCameraReverseGeo": null,
"reeferFaultDetails": null,
"reeferZoneTemperatureDetails": null,
"reeferLatitude": null,
```

```

    "reeferLongitude": null,
    "reeferReverseGeo": null,
    "reeferStatusDetails": null,
    "reeferBatteryDetails": null,
    "reeferFuelLevelDetails": null,
    "reeferHoursDetails": null,
    "reeferTemperatureSensorsDetails": null
  },
  {
    "id": "e309c2e6-0c26-4971-a452-808b8bf5fbe9",
    "rawMessageId": "ae0e9548-0b32-41a5-ac00-1bd11491a8ae",
    "recorded": "2024-07-18T21:26:51Z",
    "sensorBoxMfgrId": 6,
    "sensorBoxId": "b64cf17f-e396-48a5-9477-9be75d4fa5c9",
    "extMfgrSerialNum": "1100240400007",
    "vehicleId": "af102ab3-b3a5-4750-a38b-67eb1840942e",
    "vin": "FPXT5300TEST00007",
    "customerId": 9,
    "customerName": "Morey Test Trailers",
    "reportTypeName": "XT5300_DATA_EVENT_CODE",
    "latitude": 34.020355,
    "longitude": -84.548035,
    "altitude": null,
    "satellitesCount": null,
    "speed": null,
    "heading": null,
    "accelerometerMotion": null,
    "lightRed": null,
    "lightGreen": null,
    "lightYellow": null,
    "lightBlack": null,
    "lightBrown": null,
    "tisStatus": null,
    "tetherStatus": null,
    "odometerAbs": null,
    "odometerGps": null,
    "batteryVoltageTractor": null,
    "batteryVoltageTrailer": null,
    "doorOpened": null,
    "bogieWeight": null,
    "loaded": null,
    "loadWeight": null,
    "created": "2024-07-18T21:27:02.452295Z",
    "unitId": null,
    "absMilesTraveled": null,
    "gpsMilesTraveled": null,
    "lightFaults": null,
    "absFaults": null,
    "brakeLiningInsufficientStatus": null,
    "amberLampStatus": null,
    "vehicleElectricalSupplyInsufficientStatus": null,
    "sensorBoxMfgrName": null,
    "reportTypeId": 71,
    "withinCustomerGeofenceNames": "marksGeoFinal",
    "reverseGeoFullAddress": "2844 Cottonwood Drive, Cobb County, GA 30144, United States of America",
    "absFaultLevelId": null,
    "absFaultDescription": null,
    "timeElapsedMs": null,
    "updated": "2024-07-18T21:27:02.452295Z",
    "durationInStateMs": null,
    "idleEventCount": null,
    "idleTimeDurationMs": null,
    "adjOdometerGps": null,
    "adjOdometerAbs": null,
    "occupiedCustomerSpecificGeofenceIds": null,
    "occupiedGeofenceIds": null,
    "occupiedGeofences": "{\"f7ce2348-a878-4430-afc1-28d335a63f15\":{\"name\":\"marksGeoFinal\"}}",
    "occupiedCustomerSpecificGeofences": "{\"f7ce2348-a878-4430-afc1-28d335a63f15\":{\"name\":\"marksGeoFinal\"}}",
    "faultLevel": null,
    "tseFaults": null,
    "backupBatteryVoltage": null,
  }

```

```
        "originalRecorded": null,
        "isRecordedCorrected": null,
        "tseFaultDetails": null,
        "wheelEndDetails": null,
        "tirePressureDetails": null,
        "reverseGeo": "{\\"place\\":\\"Cobb County\\",\\"region\\":\\"Georgia\\",\\"address\\":\\"2844 Cotto",
        "absSeverityLevel": null,
        "thermoKingDetails": null,
        "ultrasonicCargoSensorVoltage": null,
        "cargoCameraImageS3Key": null,
        "cargoCameraFullness": "Failed",
        "thermoKingAlarmCodes": null,
        "cargoCameraLat": null,
        "cargoCameraLong": null,
        "cargoCameraReverseGeo": null,
        "reeferFaultDetails": null,
        "reeferZoneTemperatureDetails": null,
        "reeferLatitude": null,
        "reeferLongitude": null,
        "reeferReverseGeo": null,
        "reeferStatusDetails": null,
        "reeferBatteryDetails": null,
        "reeferFuelLevelDetails": null,
        "reeferHoursDetails": null,
        "reeferTemperatureSensorsDetails": null
    }
}
}
```