# TIKD DeLorean API

The TIKD RESTful API is entirely event based allowing you to sit back while we monitor for, match, and settle citations received by fleet vehicles. There are four types of messages that you will send to TIKD: subscribe (/fleets), unsubscribe (/fleets), service started (/renters), service ended (/renters).

TIKD hosts a unique fleet repository for all partner fleets. In order to keep your fleet repository up-to-date, you must notify TIKD of all vehicles that are

1. subscribed, and
2. unsubscribed

from your platform.

TIKD begins monitoring once service has been completed. In order for TIKD to properly determine liability of a citation, we must be sent a message when:

3. the service has started
4. the service has ended

TIKD needs renter information in order to transfer liability. TIKD must also be provided with a `rentalId` on both `'service-started'` and `'service-ended'` events, so that we can tie a `'service-ended'` event back to the appropriate `'service-started'` event.

**Base URL Endpoints**

| | |
|---|---|
| **Testing** | https://rjbsapa8a9.execute-api.us-east-1.amazonaws.com/dev |
| **Production** | https://v2ckihx5jk.execute-api.us-east-1.amazonaws.com/prod/ |

**Authentication**

Authenticate your account by including your Authentication token, provided by TIKD.

Your API key carries many privileges, so make sure to keep them secure. Don't share them in publicly accessible areas such as GitHub, client-side code, etc. All API requests must be made over HTTPS. Calls made over plain HTTP and calls made without authentication will fail.

**Errors**

TIKD uses conventional HTTP response codes to convey the success or failure of an API call. Codes in the 200 range indicate success; codes in the 400 range indicate an error based on the information provided; codes in the 500 range indicate an error on TIKD's servers.

| | |
|---|---|
| **200** OK | The request has succeeded. |
| **400** Bad Request | The request could not be understood by the server due to bad syntax. |
| **401** Unauthorized | The request requires a valid API key. |
| **403** Forbidden | The server understood the request but is refusing to fulfill it. |
| **404** Not Found | The requested resource could not be found. |
| **409** Conflict | The request has a duplicate parameter. |
| **500** Internal Service Error | There is an unexpected error on TIKD's end. |

# TIKD Subscription Quickstart

Quickly and securely subscribe your fleet and send rental events.

Subscribing your fleet with TIKD is simple two-step process that includes subscription of vehicles and passing rental events to TIKD.

**Step 1: Subscribe your fleet to TIKD**

All vehicles actively registered on your platform should be subscribed with TIKD for citation monitoring, by sending a `'subscribe'` POST request to TIKD's /fleets endpoint with the vehicle information payload. When vehicles are removed from your system, an `'unsubscribe'` event should, likewise, be sent to TIKD.

**Endpoint:**

POST
/fleets

**Headers:**

"Accept": application.vnd.fleets.v1+json
"Authorization": token provided by TIKD

**Body:**

```
{
      "transactionId": "(String) unique ID of the transaction generated by
      partner",
        "eventName": "(String) 'subscribe' OR 'unsubscribe'",
        "vehicleInfo": {
                "plateNumber": "(String - Alphanumeric) License plate of
                Vehicle",
                "plateState": "(String - two characters) Tag State of
                Vehicle",
                "vin": "(String) Vehicle Identification Number",
                "metroArea": "(String) The area where the Vehicle will be
                driven",
                "ownerInfo":    {
                        "email": "(String) Contact e-mail of Owner"
                }
        }
}
```

**Response:**

```
{
      "transactionId": "(String) acknowledgement of transaction",
}
```

**Step 2: Send rental events to TIKD**

Once a fleet is subscribed, TIKD monitors for citations based on “rental periods”. TIKD should be informed when a rental period has started, with a `'service-started'` event and also when a rental period has finished with a `'service-ended'` event. Once we get the `'service-ended'` event, we will begin to monitor the vehicle for any citations incurred during the rental service.

| | |
|---|---|
| `address` | Renter `address` information is necessary to transfer the liability of the citations |

**Endpoint:**

POST

/renters

**Headers:**

“Accept”: application.vnd.renters.v1+json

“Authorization”: token provided by TIKD

**Body:**

```
{
      "rentalId": "(String) unique ID of a particular rental period",
      "eventName": "(String) 'service-started' or 'service-ended'",
      "transactionDate": "date and time the event occurred (ISO 8601 UTC)",
      "rentalVehicle": {
               "plateNumber": "(String - Alphanumeric) License plate number
               of Vehicle",
               "plateState": "(String - two characters) Tag State of Vehicle"
        },
        "renterInfo": {
               "email": "(String) Contact e-mail of Renter",
               "firstName": "(String) First name of Renter",
               "lastName": "(String) Last name of Renter",
               "licenseNumber": "(String) Driver's License Number of Renter",
               "licenseStateIssued": "(String) Driver's License Issue State",
               "address": {
                        "street1": "(String) Street Address Line 1",
                        "street2": "(String) Street Address Line 2",
                        "city": "(String) City",
                        "state": "(String) State",
                        "zip": "(String) Zip/Postal Code"
               }
        }
}
```

**Response:**

```
{
      "transactionId": "(String) acknowledgement of transaction",
}
```

Questions? Get help now from our support team at enterprise@tikd.com.