
Carvoyant API Documentation

Release 1.0

Matt Galvin, Ronnie Williams

January 08, 2016

1	Getting Started	1
1.1	Getting Started	1
2	API Overview	13
2.1	API Overview	13
3	API Reference	17
3.1	API Reference	17
4	Sandbox API	63
4.1	Sandbox API	63
5	Driver Dashboard	67
5.1	Driver Dashboard	67
6	Sample Integrations	77
6.1	Sample Integrations	77

Getting Started

1.1 Getting Started

1.1.1 OAuth2 / Delegated Access

The Carvoyant system uses an OAuth2 implementation in order to allow users to delegate access to the applications that they want to use. All four of the [Authorization Grant types](#) have been implemented, however, only Authorization Code, Implicit and Client Credentials are available to our standard development partners. If you need to support the Resource Owner type, please contact us to discuss.

Here is a brief summary of how you will use each of the grant types when accessing the Carvoyant system. Please refer to the specification for details on each grant type.

Grant Type	Usage
Authorization Code	This type will be used in the scenario where the system making the actual API call is in a secured environment. For example, web applications where the Carvoyant API call originates from our partners server will use this method. The development partner will direct the user to a Carvoyant hosted web page where the user will log in and grant access to the development partner. The Carvoyant system will redirect the user back to an endpoint that the development partner hosts which will contain an authorization code. The development partner will then make a request from their controlled environment to retrieve an access token to the users Carvoyant data. That call will be a server to server call over SSL and will require the development partners credentials. The response will include an access token and a refresh token per the OAuth2 specification. This is the preferred grant type as it provides the highest level of security in the token negotiation process.
Implicit	This type will be used when the calls into the Carvoyant system are made from an insecure environment. The primary example for this use case is a mobile application. In this process, the development partners application (typically running on a resource controlled by the user) directs the user to a Carvoyant hosted web page where the user will log in and grant access to the development partner. The Carvoyant system will redirect the user back to an endpoint that the development partner specifies. That redirect will contain the access token per the OAuth2 specification. In this grant type, refresh tokens are not provided.
Client Credentials	There are a few calls into the Carvoyant API that the development partner will make that does not require a user access token. A primary example of this is the call to register a new account. Since the account does not exist yet, there is no user to grant access. API calls using this grant type will still expect an access token, but that token will be requested using only the development partners credentials.

Warning: Configuration Requirements

In order to ensure the security of the OAuth2 process, there are two requirements that must be met. When you registered your application, one of the fields that you were asked for was the “Register Callback URL”. This is the URL that the OAuth2 process will redirect users back to after they authorize your application. Specifically, it is the value of the `redirect_uri` in all of the below examples. There are two important aspects to this

1. The value **MUST** be an SSL protected URL of the form `https://...` and the certificate must be signed by a trusted certificate authority (ie, self signed certificates are not supported).
2. All requests that are made into the OAuth2 process must use that value for the `redirect_uri`. Mismatched values will generate errors and will result in an access token not being generated.

If you need to change this value, or are unsure of what it is set to, you can access it by logging into the [Developer Console](#) and selecting “My Account” in the top right. This will let you access any applications that you have created keys for.

Examples

Below are example calls using cURL to show how to generate tokens under each grant type.

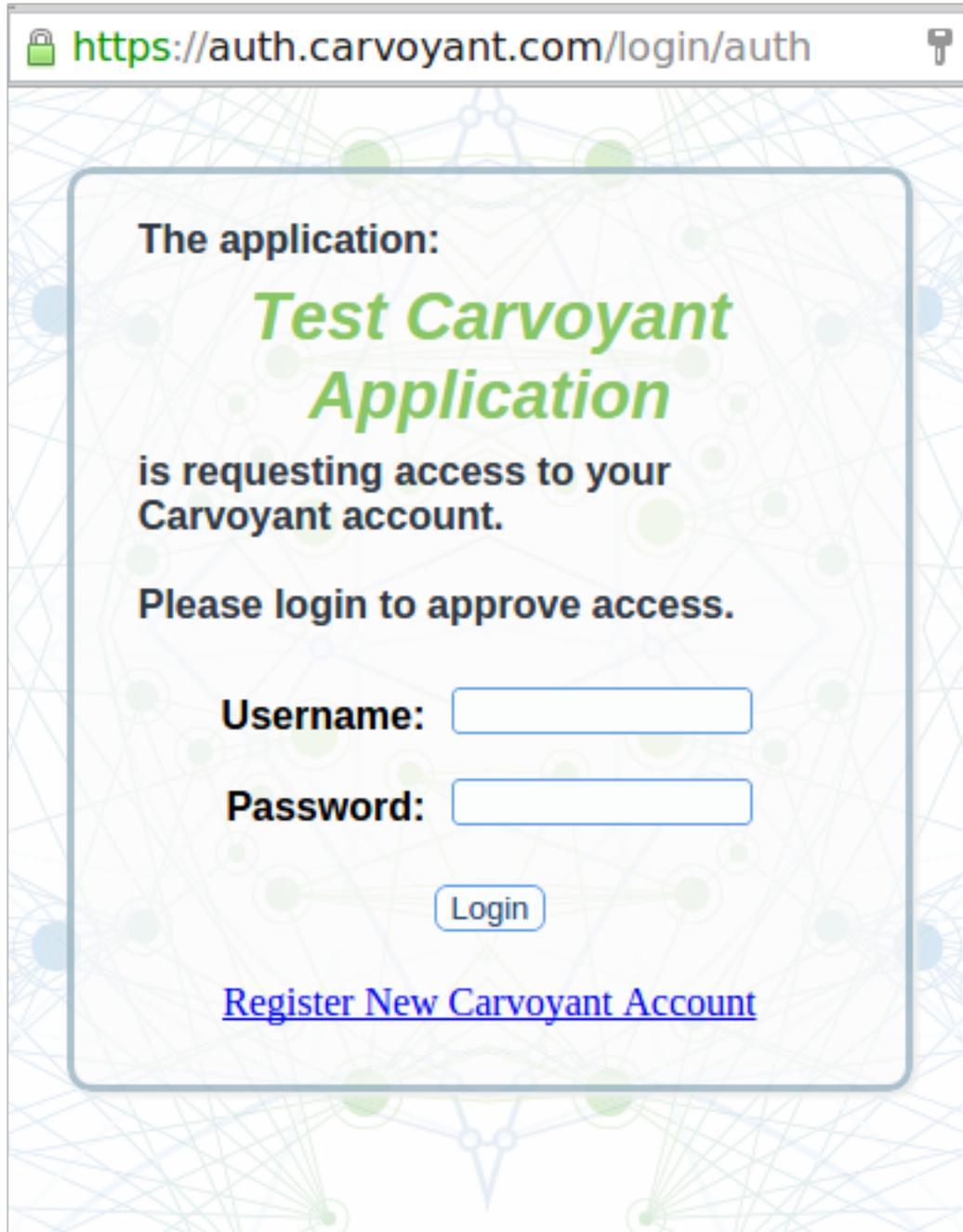
Authorization Code

The first step is to request the authorization from the user. This call will return a redirect to the login page.

Authorize Request:

```
curl -i -X GET -d "client_id=pekfaf6jxk6suyXXXXXXXXXX" --data-urlencode "redirect_uri=http://test.carvoyant.com"
HTTP/1.1 302 Found
Content-Type: text/plain; charset=UTF-8
Date: Mon, 28 Apr 2014 14:03:51 GMT
Location: https://auth.carvoyant.com/login/auth
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=2D3C7E5AA412DF98124B8AC7121FEF7D; Path=/; Secure; HttpOnly
Content-Length: 0
Connection: keep-alive
```

If you did this through a browser, the login screen would appear to the user:



<https://auth.carvoyant.com/login/auth>

The application:
Test Carvoyant Application
is requesting access to your Carvoyant account.
Please login to approve access.

Username:

Password:

Login

[Register New Carvoyant Account](#)

After logging in, the user will be redirected to the `redirect_url` specified in the authorization request with the authorization code appended to it. For the request above, the redirect uri would look like::

```
http://test.carvoyant.com/?code=v369uars628mgkXXXXXXXXXX
```

From this response, the server running at `test.carvoyant.com` would parse the authorization code and make a request to the token endpoint to read an actual access token. Note that this request requires the client id and secret key for the development partner to be passed as the Basic Authentication credentials.

Access Token Request:

```
curl -i --user pekfaf6jxk6suyXXXXXXXXXX:XXXXXXXXXX -d "grant_type=authorization_code" -d "code=v369ua"
```

The response will include a json body with the access token information.

Access Token Response:

```
HTTP/1.1 200 OK
Cache-Control: no-store
Content-Type: application/json;charset=UTF-8
Date: Mon, 28 Apr 2014 14:24:34 GMT
Server: Mashery Proxy
X-Mashery-Responder: prod-j-worker-us-east-1c-31.mashery.com
Content-Length: 161
Connection: keep-alive
{
  "token_type": "bearer",
  "mapi": "pekfaf6jxk6suyXXXXXXXXXX",
  "access_token": "dmnda67wbdnyayXXXXXXXXXX",
  "expires_in": 86400,
  "refresh_token": "f2hques6fpg37d2XXXXXXXXXX"
}
```

At this point, the development partners system would store the access token and refresh token and use them for future requests.

Implicit

The first step is to request the authorization from the user. This call will return a redirect to the login page.

Authorize Request:

```
curl -i -X GET -d "client_id=pekfaf6jxk6suyXXXXXXXXXX" --data-urlencode "redirect_uri=http://test.carvoyant.com"

HTTP/1.1 302 Found
Content-Type: text/plain; charset=UTF-8
Date: Mon, 28 Apr 2014 14:03:51 GMT
Location: https://auth.carvoyant.com/login/auth
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=2D3C7E5AA412DF98124B8AC7121FEF7D; Path=/; Secure; HttpOnly
Content-Length: 0
Connection: keep-alive
```

If you did this through a browser, the login screen would appear to the user:

<https://auth.carvoyant.com/login/auth>

The application:

Test Carvoyant Application

is requesting access to your Carvoyant account.

Please login to approve access.

Username:

Password:

Login

[Register New Carvoyant Account](#)

After logging in, the user will be redirected to the `redirect_url` specified in the authorization request with the access token appended to it. For the request above, the redirect uri would look like::

```
http://test.carvoyant.com/?access_token=2pr9tvk3vgnr9aXXXXXXXXXX&token_type=bearer&expires_in=86400
```

From this response, the endpoint at `test.carvoyant.com` would parse the access token and begin making API calls.

Client Credentials

In this grant type, there is no user interaction. The development partner requests a token for their client credentials directly.

Access Token Request:

```
curl -i --user pekfaf6jxk6suyXXXXXXXXXX:XXXXXXXXXX -d "grant_type=client_credentials" "https://api.c
```

The response will include a json body with the access token information.

Access Token Response:

```
HTTP/1.1 200 OK
Cache-Control: no-store
Content-Type: application/json;charset=UTF-8
Date: Mon, 28 Apr 2014 14:57:20 GMT
Server: Mashery Proxy
X-Mashery-Responder: prod-j-worker-us-east-1d-32.mashery.com
Content-Length: 161
Connection: keep-alive
{
  "token_type": "bearer",
  "mapi": "pekfaf6jxk6suyXXXXXXXXXX",
  "access_token": "n45u7eufgdmfysXXXXXXXXXX",
  "expires_in": 86400,
  "refresh_token": "r9acw76k327g3cXXXXXXXXXX"
}
```

At this point, the development partners system would store the access token and refresh token and use them for future requests.

Using the Access Token

In all grants types, once the access token has been retrieved, calls into the Carvoyant system are done by passing the access token in the HTTP request within the Authorization header. For example (using a sample access token from the Authorization Code grant type):

Request:

```
curl -i -H "Authorization: Bearer dmnda67wbdnyayXXXXXXXXXX" "https://api.carvoyant.com/v1/api/vehicle
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: Mon, 28 Apr 2014 15:06:47 GMT
Server: Apache-Coyote/1.1
X-Mashery-Responder: prod-j-worker-us-east-1b-36.mashery.com
Content-Length: 1205
Connection: keep-alive
{"vehicle": [{"name": "1999 Jeep Wrangler SE", "label": "Custom dune buggy", "vehicleId": 3, "deviceId": "C2
```

Using the Refresh Token

When using the Authorization Code grant type, you will receive a refresh token in addition to the access token. Any time up until the expiration of the refresh token (currently 30 days past the access token expiration), you can generate

a new access token without requiring the user to go through the authorization process.

1.1.2 Access & Registration Workflows

When a partner creates an application for the Carvoyant platform, it is necessary to handle two different user scenarios. In the first scenario, the end user does not already have a Carvoyant account and needs to be registered with the Carvoyant platform. In the second scenario, the user already has a Carvoyant account and simply wants to connect the partners application to their account. This poses some questions regarding how the Carvoyant account is created (when necessary), how Carvoyant validates that the user has granted access to the partner application and how the partner uses that access. If you are not already familiar with the OAuth mechanism used in the Carvoyant Platform, please read that first.

Scenario 1 - New Carvoyant Account

In many cases, our partners will be bringing on new Carvoyant users to our platform as a part of their application. Perhaps you are creating a brand new connected car application. Or maybe you're simply adding connected car features to your existing application. In either case, it is necessary to create the Carvoyant user account. This can be done two ways. First, you can just treat the "new" and "existing" customers the exact same way. You simply prompt them to authorize Carvoyant to access their account and then take them to our authentication server as outlined in our OAuth documentation. If you want to use this method, please see Scenario 2.

If you don't want to require the user to go through the Carvoyant UI for account creation, you can simply call the account creation API endpoint in our system. You will need to pass over all of the required fields for a new account. The response to the account creation call will include an authorization code. That code can then be used to programmatically generate the access token. An example request flow looks like this:

Account Creation Request:

```
curl -X POST -H 'Authorization: Bearer m7dwthfgv9dvdpxXXXXXXXXXX' -H 'Content-Type: application/json'
```

Account Creation Response:

```
{
  "account": {
    "id": 166,
    "firstName": "Doc",
    "lastName": "Test",
    "username": null,
    "dateCreated": "20141112T195503+0000",
    "email": "matt@carvoyant.com",
    "zipcode": null,
    "phone": null,
    "timeZone": null,
    "preferredContact": null,
    "accessToken": {
      "code": "dj6fpxdbkh3xqyXXXXXXXXXX"
    }
  },
  "totalRecords": 1
}
```

Note the `accessToken.code` in the response.

Now lets get the access token:

Access Token Request:

```
curl --user r6dwmz2zxxqms7sac2r8mdqa:XXXXXXXXXX -d "grant_type=authorization_i=https://test.carvoyant.com" --url https://api.carvoyant.com/oauth2/token
```

Access Token Response:

```
{
  "access_token": "6dpmu7c4gv3s3yXXXXXXXXXX",
  "token_type": "bearer",
  "expires_in": 86400,
  "refresh_token": "z9h69fqx4cdkfjXXXXXXXXXX"
}
```

Now you've successfully registered a new Carvoyant account and generated an access token into the API for that user without requiring the user manually log in.

Note: Accounts created in this way make it very easy for the end user to misunderstand who is collecting their data and who is storing it. If you are generating accounts and access tokens for Carvoyant users in this manner, you must notify them of the following:

1. Their Carvoyant user credentials (ie, the username and password that was passed in to the account creation).
2. A link to <https://driver.carvoyant.com> where the user can access their Carvoyant account.
3. A clear statement that their connected car data is being collected on their behalf by Carvoyant and made available to you, our partner.

If there are any questions about this, please contact us: [contact us](#).

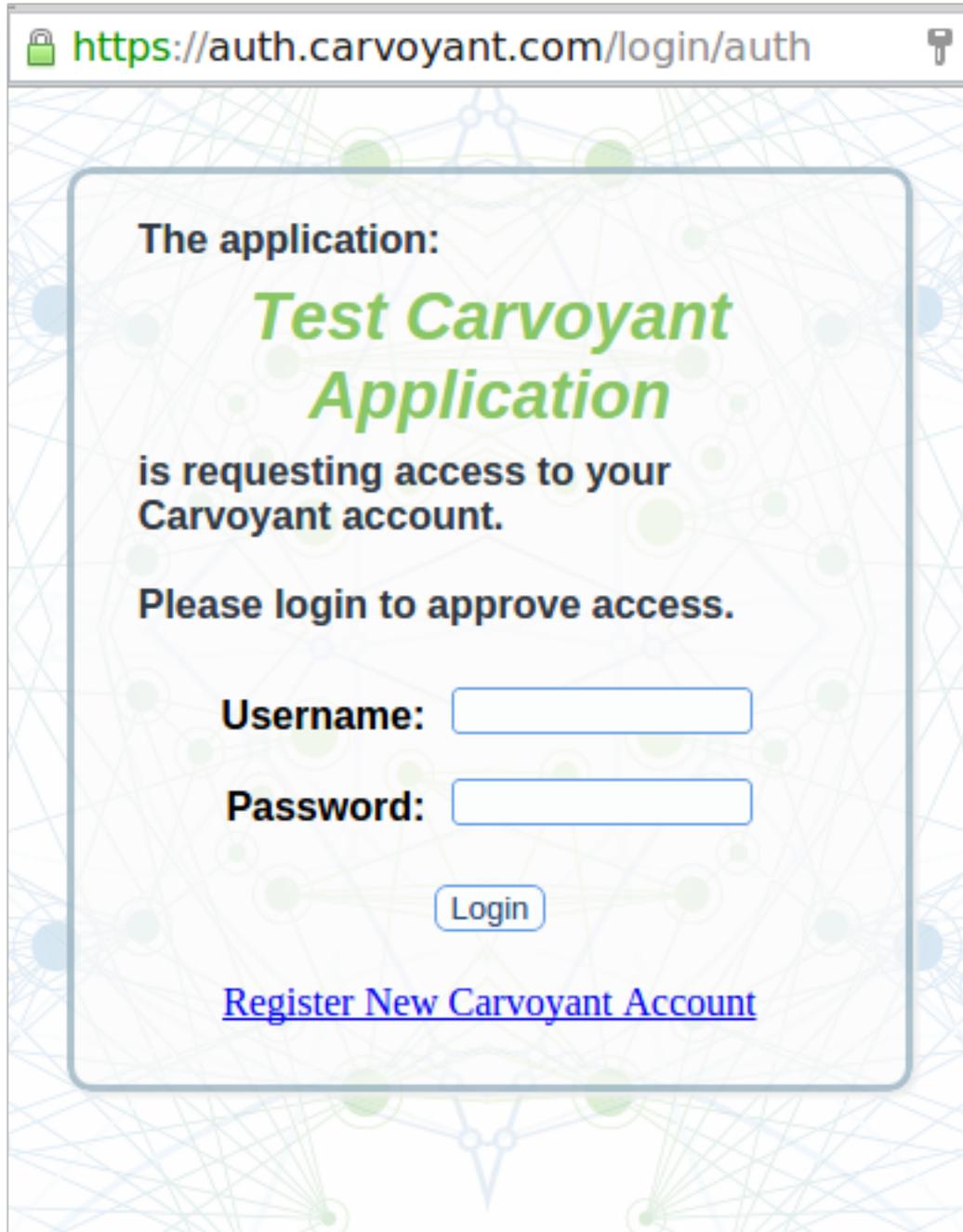
Scenario 2 - Existing Carvoyant User (or using the Carvoyant new account UI)

As the Carvoyant user base grows, connected users are going to want to connect more and more applications to their connected car. In order to do this, application providers need to allow those users to authorize the partner's application to the user's Carvoyant account. This is done by using our authorization server. Note that this process is also outlined on our [OAuth2 / Delegated Access](#) documentation.

When the you are ready to have your user authorize you to their Carvoyant account, you simply open a browser and redirect them to our authorization page. The URL looks like this::

```
https://sandbox-auth.carvoyant.com/OAuth/authorize?client_id=r6dwmz2zxxqms7XXXXXXXXXX&redirect_uri=https://example.com/callback
```

Replace the `client_id` with your applications `client_id` and the `redirect_uri` with the URL that you will listen on for the response. The user will be presented with a screen like this:



<https://auth.carvoyant.com/login/auth>

The application:
Test Carvoyant Application
is requesting access to your Carvoyant account.
Please login to approve access.

Username:

Password:

Login

[Register New Carvoyant Account](#)

Note that the user can log in with their existing credentials or they can register a new one right there. From a partner's perspective, you can simply provide a "Connect Carvoyant" action and this will solve the integration for both new Carvoyant users and already existing Carvoyant users. After the user logs in and authorizes access, they will be redirected back to the URL you specified in the `redirect_uri` parameter and it will include the authorization code. The URL will look like this::

```
https://test.carvoyant.com?code=dj6fpxdbkh3xqyXXXXXXXXXX
```

The code query parameter in that URL is the authorization code that is used to request the access token. That process is identical to what is outlined above. Note that this example is using the authorization code grant type. The same process can be followed for the `implicit` grant type.

1.1.3 Overview

Before accessing the Carvoyant system as a developer, you must first register on our developer portal. After your account is registered, you will be able to make calls into our system. One of our main philosophies is that our users have full control over who can access the data for their vehicles. In order to facilitate this, nearly all calls that a developer makes into the system requires an [OAuth2](#) access token that the end user has granted. Once a token has been granted, the various Carvoyant resources will be accessible through the API.

1.1.4 Developer Registration

The first step in building an application for the Carvoyant system is to register with us as a developer. We have several different developer levels depending on your interest. You can view the different developer levels and associated pricing [here](#) or you can just go register [here](#).

During registration you will enter in some basic information about your application and you will be given an API key to use in your application. Please only create a Standard Developer key if you are also purchasing a developer device or already have a Carvoyant enable vehicle. A free Trial key is available if you are just taking a look.

1.1.5 Sandbox Access

In order to allow developers to try out our system, or even just develop against a non-production environment, we've created a Sandbox version of our API. This environment matches our production environment with two important distinctions. First, it is not possible to connect a live vehicle to an account in this system. All of the data in this system is "fake". Second, all developers have the ability to call an endpoint that allows them to create their own vehicle data. In this way, you can programmatically generate the vehicle data that you need to test your application. Review the information about the Sandbox API to get started.

1.1.6 Paid Standard Developer Accounts

Every paid Standard Developer account also includes one telematics device to connect your car to the Carvoyant platform. Before that the data for your car can be accessed, you must register as a Carvoyant user [here](#) (this can also be done through an API call). It's important to recognize the difference between a developer account and a user account. Developer accounts are authorized to make API calls against the Carvoyant platform.

1.1.7 Developer vs. User Accounts

When working with the Carvoyant platform, it's very important to understand the difference between Developer accounts and User accounts. A Developer account is registered through <https://developer.carvoyant.com> and is what you will use to interact with the Carvoyant API. A User account is registered with the Carvoyant platform (either programmatically through the /account endpoint, or on the driver dashboard at <https://driver.carvoyant.com>). User accounts contain the vehicles which contain the data collected by the telematics device.

All Carvoyant Users can create a Developer account to access their own connected car data.

1.1.8 Making an API Call

The best way to get familiar with our API is to jump in and start using it. Using our [interactive documentation](#), you can start calling the API directly.

We've also added a couple of sample applications to our GitHub repository.

[Car-Locator](#) is an Android application that uses the Implicit grant type to authenticate a user. Once logged in, it displays a map of the users Carvoyant enabled vehicles.

[Example-Carvoyant-Web-App](#) is a Grails web application that uses it's own security model to authenticate it's application users and then lets that user connect to their Carvoyant account. This application uses the Authorization Code grant type.

API Overview

2.1 API Overview

The Carvoyant API is a RESTful web API that allows our customers and partners to access Carvoyant data programmatically. All API calls must be made over SSL and access into the system is controlled by an OAuth2 implementation. The URI to access the API is <https://api.carvoyant.com/v1/api>. Every request is expected to include the Authorization header containing the bearer token generated through one of the OAuth2 grant types. See [OAuth2 / Delegated Access](#) for details on how to request an access token.

In following along with the standard approach for a RESTful service, our API is structured in terms of its Resources. So you won't see "actions" listed in here. Instead, you will find the information organized in terms of the resources that the Carvoyant API exposes. For each resource, you will find a description of the object as well as the various requests that can be made against it (ie, GET, PUT, POST, DELETE).

2.1.1 JSON Success Response Format

Responses from the Carvoyant system will follow a common structure for all API calls. The general format will be::

```
{
  "<object name>":
    [
      {<object 1>},
      {<object 2>},
      {<object 3>}
    ],
  "totalRecords": <total records>,
  "actions":
    [
      {
        "name": "<action1 name>",
        "uri": "<action1 uri>"
      },
      {
        "name": "<action2 name>",
        "uri": "<action2 uri>"
      }
    ]
}
```

<object name> will be defined by the specific API call that is made. In cases where a list is returned, the object value will be an array. If a single object is returned then the object value will just be the object (it will not be a single

element array). The “actions” array will include any subsequent API calls that can be made related to the returned object (this is our first pass at implementing [HATEOAS](#)). We decided to standardize the payload format rather than rely on something like the HTTP Link header for passing back available actions.

Note that the `totalRecords` property will only be present when the object returned is a list. The value of `totalRecords` will be the total count of records contained within the Carvoyant system. It will not be the number of records returned in the response. This will be useful with returned data that is paged. For instance, the system might have 115 records stored but the pagination parameters may only return 20 records in this particular response. In that scenario, the `totalRecords` value will be 115.

2.1.2 JSON Error Response Format

API calls that result in an error will have a different format than successful message. The first difference is that the HTTP response code will not be 200. It will be an HTTP code more reflective of the error encountered. For example, a 404 will be returned if a resource that does not exist is requested. The body of the response will contain the following JSON response format::

```
{
  "error":
  {
    "httpStatus": <http status code>,
    "detail": "<string message>"
  }
}
```

2.1.3 HTTP Verbs

One of the recurring areas of confusion around APIs that strive to be RESTful has to do with the mapping of HTTP verbs to actual server side actions. Here are some guidelines for how the Carvoyant API handles the various verbs for different resource types.

Verb	Container (e.g. /vehicle)	Entity (e.g. /vehicle/{id})
GET	Returns a list of all entities in the container.	Returns the specified entity.
PUT	Currently not supported for any container type. In practice, this would replace the entire container with the entities specified in the request.	Replaces the specified entity.
POST	Creates a new entity within the container.	Updates the specified entity. Only data elements included in the request will be updated. Non-specified fields will not be changed.
DELETE	Currently not supported for any container type. In practice, this would delete all entities in the container.	Deletes the specified entity.

2.1.4 Sorting and Paging

Many of the calls within the Carvoyant API return sets of data. In some cases, very large sets of data. For a number of reasons, it would not be feasible return the entire data set at once (we have some vehicles with over 100,000 data points after only 4-5 months of being on our system). In order to address this, we have standardized the mechanism to sort and page results. Any API call that supports sorting or pagination will do so as described here.

Sorting

In most cases, the objects being listed will sort based on a timesamp. It might be the time that the trip was started or the time a data point was collected. The documentation for the individual API call will indicate how the results are sorted. In order to control the sort order of that field, you can include the following field in your query string:

`sortOrder=asc` will return the results in ascending order

`sortOrder=desc` will return the results in descending order

Paging

Paging is the mechanism by which the number of results returned are limited and the ability to retrieve the “next” set of data in the list. There are two pieces of data important in specifying a paging mechanism. The size of the data set returned and the starting point in the full data set where the returned results will begin.

`searchLimit=<x>` will specify how many results are included in the response

`searchOffset=<y>` will specify the offset from the beginning of the data set that is returned

Generally, you will not need to figure out the values for the paging links manually. They will be specified in the actions array of the return as “next” and “previous”.

API Reference

3.1 API Reference

3.1.1 Data Types

There are several defined data types beyond the normal primitives that are used within the Carvoyant API. They are described here.

DataKey

A DataKey is a string that represents a type of data in the Carvoyant system.

Key String	Description
GEN_DTC	Diagnostic Trouble Codes
GEN_VOLTAGE	Battery Voltage
GEN_TRIP_MILEAGE	Trip Mileage (calculate from ignition on to ignition off via GPS)
GEN_ODOMETER	Vehicle Reported Odometer
GEN_WAYPOINT	GPS Location
GEN_HEADING	Heading (degrees clockwise from due north)
GEN_RPM	Engine Speed
GEN_FUELLEVEL	Percentage of Fuel Remaining
GEN_FUELRATE	Rate of Fuel Consumption
GEN_ENGINE_COOLANT_TEMP	Engine Temperature
GEN_SPEED	Maximum Speed Recorded (since the previous reading)
GEN_NEAREST_ADDRESS	The physical address nearest to the associated Waypoint
VEHICLE_EVENT_CONNECTED	Vehicle device was connected
VEHICLE_EVENT_DISCONNECTED	Vehicle device was disconnected
VEHICLE_EVENT_TOWED	Towing has been detected
VEHICLE_EVENT_TOWED_BREADCRUMB	A breadcrumb indicating the vehicle is being towed
VEHICLE_EVENT_HARSH_ACCEL	Harsh acceleration
VEHICLE_EVENT_HARSH_DECEL	Harsh deceleration
VEHICLE_EVENT_HARSH_RIGHT	Harsh right turn
VEHICLE_EVENT_HARSH_LEFT	Harsh left turn
VEHICLE_EVENT_HARSH_IMPACT	Impact has been detected

DateTime

All timestamps will be returned in the ISO 8601 standard format (yyyyMMddTHHmmsZ).

- yyyy : 4 digit year
- MM : 2 digit month
- dd : 2 digit day
- HH : hour (on a 24 hour scale)
- mm : minutes
- ss : seconds
- Z : timezone offset

Implementations should honor the timezone but in most cases all timestamps will be returned in UTC with a timezone offset of +0000 (we expect and use the +0000 offset for UTC rather than the Z shorthand that the spec provides for).

For example, the string 20130526T204840+0000 denotes May 26, 2013 at 8:48 and 40 seconds, PM, in UTC.

NotificationPeriod

NotificationPeriod is a string that represents when EventNotifications are sent from the Carvoyant system.

Key String	Description
CONTINUOUS	Every time a data point is collected and the EventSubscription criteria is met, an EventNotification will be sent. When the criteria is no longer met, an EventNotification will NOT be sent.
STATE-CHANGE	An EventNotification will be sent when the EventSubscription criteria is first met. Another EventNotification will be sent when the criteria is no longer met.
INITIAL-STATE	An EventNotification will be sent when the EventSubscription criteria is first met. Another EventNotification will not be sent until the criteria is no longer met, and then is met again.
ONE-TIME	An EventNotification will be sent when the EventSubscription criteria is first met and the EventSubscription will be deleted. No further event EventNotifications will be sent.

Here's a chart to help visualize when event notifications are sent for each state:

State	Collection Time			
	08:00 AM	08:05 AM	08:10 AM	08:15 AM
	Event Starts			Event Ends
CONTINUOUS	X	X	X	
STATECHANGE	X			X
INITIALSTATE	X			
ONETIME	X			

Note: Not all NotificationPeriod values will be supported for every EventNotification

Waypoint

A waypoint is a GPS location at a specific point in time.

Properties

Property Name	Type	Description	Optional
timestamp	DateTime	The time that the location was recorded	N
latitude	Float	The latitude of the location	N
longitude	Float	The longitude of the location	N

JSON Sample:

```
{
  "timestamp": "20140116T152440+0000",
  "latitude": 28.088445,
  "longitude": -82.578451
}
```

EventType

An EventType represents an event that can be triggered within the Carvoyant system. API users can subscribe to these events using an [EventSubscription](#). This page describes the details of each event type. In the details below you will find the following:

- **Event Scope:** Events can be set at the system, account or vehicle scope.

Scope	Description	Query Path Prefixes
System	These subscriptions are created using the Client Credentials mechanism and are not account specific. They allow you to subscribe to system level events.	/system
Account	These subscriptions are applicable to the account that the access token applies to.	/account
Vehicle	These subscriptions are applicable to a specific vehicle. They can be specified at the account level in which case the Carvoyant system will ensure that all vehicles on your account have the subscription created for it.	/account /vehicle

- **Event Type Key:** This is the string that will be used in [EventSubscription](#) and [EventNotification](#) objects wherever you see {event-type}.
- **Subscription Properties:** This will be the list of properties that are specified when the subscription is created.
- **Supported Notification Periods:** This will list which [NotificationPeriod](#) values are supported for the event type.

Geo Fence

GeoFence events will trigger when a vehicle's location is reported at a certain location. This is evaluated any time a [Waypoint](#) is received by the system.

Scope: Vehicle

Event Type Key: GEOFENCE

Subscription Properties

Name	Type	Description	Required for Creation
origin	Waypoint	The reference location on which to base event notification	Required
radius	Float	The length of the radius extending outward from the origin that creates the GeoFence boundary. This is specified in miles.	Required
ignition-Status	String Enumeration: ON OFF RUNNING ANY	Defines the required ignition state of the vehicle that must be met in order for the event notification to occur.	Required
bound-aryCon- dition	String Enumeration: INSIDE OUTSIDE	Specifies whether to trigger the event notification if the vehicle is “INSIDE” or “OUTSIDE” of the defined boundary.	Required

Supported Notification Periods

- CONTINUOUS
- STATECHANGE
- INITIALSTATE
- ONETIME

Ignition Status

Ignition status events will trigger when a vehicles engine state (ie, ignition state) is turned on or off.

Scope: Vehicle

Event Type Key: IGNITIONSTATUS

Subscription Properties: No additional properties

Supported Notification Periods

- STATECHANGE
- INITIALSTATE
- ONETIME

Low Battery

The low battery event will trigger when the voltage read from the vehicle falls below 12.0V. For other values, see the Numeric Data Keys event type.

Scope: Vehicle

Event Type Key: LOWBATTERY

Subscription Properties: No additional properties

Supported Notification Periods

- CONTINUOUS
- STATECHANGE
- INITIALSTATE
- ONETIME

Numeric Data Keys

Numeric data key events will trigger when the value of the specified [DataKey](#) meets the criteria. This can be used to customize events off of any numerical data point collected by Carvoyant.

Scope: Vehicle

Event Type Key: NUMERICDATAKEY

Subscription Properties

Name	Type	Description	Required for Creation
dataKey	DataKey	The DataKey to check against. Note that the following keys are supported: GEN_VOLTAGE, GEN_TRIP_MILEAGE, GEN_ODOMETER, GEN_HEADING, GEN_RPM, GEN_FUELLEVEL, GEN_FUELRATE, GEN_ENGINE_COOLANT_TEMP, GEN_SPEED	Required
threshold-Value	Float	The value that determines when to send the event notification in reference to the corresponding vehicle data.	Required
relationship	String: ABOVE BELOW EQUALTO	Defines the condition that is used to compare the value of the subscription against current vehicle data.	Required

Supported Notification Periods

- CONTINUOUS
- STATECHANGE
- INITIALSTATE
- ONETIME

Time Of Day

The TimeOfDay Subscription allows event notification when a vehicle is operated outside of a defined time period.

Scope: Vehicle

Event Type Key: TIMEOFDAY

Subscription Properties

Name	Type	Description	Required for Creation
start-Time	String in HH:MM format	The time of day that the vehicle is permitted to run.	Required
end-Time	String in HH:MM format	The time of day when the vehicle is no longer permitted to run.	Required
daysOfWeek	Array of String Enumeration: SUN, MON, TUE, WED, THU, FRI, SAT, SUN	Represents the days of the week that the vehicle is permitted to run.	Required
ignition-Status	String Enumeration: ON OFF RUNNING ANY	Defines the required ignition state of the vehicle that must be met in order for the event notification to occur.	Required

Supported Notification Periods

- CONTINUOUS
- STATECHANGE
- INITIALSTATE
- ONETIME

Trouble Code

The trouble code event will trigger when the vehicle reports a Diagnostic Trouble Code (DTC).

Scope: Vehicle

Event Type Key: TROUBLECODE

Subscription Properties: No additional properties

Supported Notification Periods

- INITIALSTATE
- ONETIME

Driver Behaviors

Driver behavior events trigger based on how the driver is driving. Each are determined using an internal accelerometer within the device in the vehicle.

Scope: Vehicle

Event Type Keys:

- VEHICLEHARSHACCEL: Indicates that a high rate of acceleration has been detected.
- VEHICLEHARSHDECEL: Indicates that a high rate of deceleration has been detected.
- VEHICLEHARSHRIGHT: Indicates that a hard right turn has been detected.
- VEHICLEHARSHLEFT: Indicates that a hard left turn has been detected.
- VEHICLEIMPACT: Indicates that an impact has been detected. Please note that currently, the act of plugging in or unplugging a device to the OBDII port while the vehicle is on may trigger this event.

Note: Driver behavior events will only be triggered while the vehicle is running. Specifically, this means that an impact that takes place while the vehicle is not running will not trigger an alert.

Subscription Properties: No additional properties

Supported Notification Periods

- INITIALSTATE
- ONETIME

Vehicle Events

Vehicle events are generally related to events that occur with the vehicle that do not have to do with driving activities.

Scope: Vehicle

Event Type Keys:

- VEHICLECONNECTED: Indicates that connectivity to the car has been established. For OBDII based cars, this means the device has been plugged in.
- VEHICLEDISCONNECTED: Indicates that connectivity to the car has been removed. For OBDII based cars, this means the device has been unplugged.
- VEHICLETOWED: Indicates that the vehicle is being towed. Specifically, this means the vehicle has moved a certain distance (currently 1500 meters) without the vehicle being turned on.

Note: VEHICLETOWED will be triggered if the device is unplugged and then plugged back in after moving the configured distance. If a device is unplugged and then plugged back in that distance away, the vehicle should be started. That will clear the towing indicator on the device.

Subscription Properties: No additional properties

Supported Notification Periods

- INITIALSTATE
- ONETIME

Vehicle Creation

These events allow you to react to the creation or deletion of a vehicle on an account.

Scope: Account

Event Type Keys:

- VEHICLECREATED: Indicates that the vehicle has been created in the system.
- VEHICLEDELETED: Indicates that the vehicle has been deleted from the system. Note that after receiving a notification that the vehicle has been deleted, you can no longer query against it.

Subscription Properties: No additional properties

Supported Notification Periods

- ONETIME
- CONTINUOUS

Account Authorization

This event will notify you of the change in access grants to an account for your client id. At this time, only revoke notifications will be sent.

Scope: System

Event Type Key:

- AUTHORIZATIONSTATUS: Indicates that the authorization status for the account has changed.

Subscription Properties: No additional properties

Supported Notification Periods

- ONETIME
- CONTINUOUS

Notification Properties

Name	Type	Description
authorizationStatus	String Enumeration GRANTED, REVOKED	The time of day that the vehicle is permitted to run.

DeliveryType

A DeliveryType represents how a notification should be sent when the subscription criteria is met. The following types are supported:

- HTTP_POST
- EMAIL

DeliveryType Properties

Name	Type	Description	Required for Creation
type	POST_HTTP or EMAIL	The type of delivery.	Required
active	boolean	Indicates if the DeliveryType is currently active. This can only be set by the Carvoyant system (it cannot be set through the API)	Unsupported
deactivationReason	String	If the system deactivates this DeliveryType, this field will indicate why.	Unsupported

HTTP_POST

This DeliveryType will uses HTTP POST to submit the [EventNotification](#) the specified URL.

HTTP_POST Properties

Name	Type	Description	Required for Creation
postUrl	String	The URL that will receive HTTPS POST notifications from the newly generated subscription. Note that only HTTPS endpoints are supported. The certificate protecting the post URL must be a valid signed certificate.	Required
post-headers	Map	A map that contains any headers that should be sent to the postUrl when a notification is generated.	Optional
oauth2	Map	A map that contains the values for OAuth2 based authentication of notifications endpoints.	Optional

OAUTH2 A DeliveryType can use OAuth2 as an authentication mechanism when sending the HTTP_POST. The following fields are required as a part of the oauth2 map.

OAUTH2 Properties

Name	Type	Description	Required for Creation
access_token	String	The access token to be included in the request.	Required
client_id	String	The OAuth2 client id in the receiving system.	Required
client_secret	String	The OAuth2 client secret in the receiving system.	Required
refresh_token	String	The OAuth2 refresh token for this access token. If a 401 is returned from a notification request the Carvoyant system will attempt to refresh the token. This will follow the format in the OAuth2 specification.	Required
redirect_uri	String	The URI to use when refreshing a token.	Required

When a notification is delivered using OAuth2, the initial HTTP_POST to the postUrl will contain an *authorization* header with the value “Bearer <access_token>”. If the delivery returns an HTTP status of 401 (Unauthorized), the Carvoyant system will attempt to refresh the token. A request will be sent to the redirect_uri following the OAuth2 refresh token specification. The request will use an http POST and will look like the following:

```
<redirect_uri>?grant_type=refresh_token&client_id=<client_id>&client_secret=<client_secret>&refresh_token=<refresh_token>
```

The expected response will be a JSON object that contains the properties “access_token” and “refresh_token”. Those values will update the delivery object and the notification will be retried.

EMAIL

This DeliveryType will send an email with the details of the [EventNotification](#) .

EMAIL Properties

Name	Type	Description	Required for Creation
recipient	String	The email address to send the notification to.	Required
subscriberSubject	String	The subject of the notification email.	Optional
subscriberContent	String	The body of the notification email.	Optional

3.1.2 Resources

A resource is an addressable object within the Carvoyant API. These are the primary objects within the Carvoyant system.

Account

The Account object represents a unique account within the Carvoyant system.

Properties

Property	Type	Description	Required for Creation
id	Integer	The internal system identifier for this vehicle.	Unsupported
firstName	String	The first name on the account.	Required
lastName	String	The last name on the account.	Required
dateCreated	Date-Time	The timestamp of when the account was created.	Unsupported
zipcode	String	The zipcode where the user primarily drives.	Required
email	String	Email address for the account.	Optional
phone	String	Phone number for the account.	Optional
timeZone	String	The timezone to use for time display.	Required
preferredContact	String	How the account owner prefers to be contacted.	Required
accessToken	String	A Bearer access token to use to access this account. Note that an access token is only returned by the creation of a new account.	Unsupported
username	String	The username for the account.	Required
password	String	The password for the account.	Required

Supported Verbs

- GET
- POST
- DELETE

GET

Returns one or more accounts.

Query Paths

- /account/
- /account/{account-id}

Query Parameters

Parameter	Description
account-id	The Carvoyant identifier of the account. If the account-id is not specified, then all accounts available will be returned. In most cases, only one account is available so both calls will return the same account.

Call Options

Sortable	No
Pageable	No

Sample JSON Response:

```
{
  "account": {
    "id": 3
    "firstName": "Speed"
    "lastName": "Racer"
    "username": "speedracer"
    "dateCreated": "20121130T144013+0000"
    "email": "speedracer@noemail.com"
    "zipcode": "33635"
    "phone": "8135551212"
    "timeZone": "America/New_York"
    "preferredContact": "PHONE"
  }
  totalRecords: null
  actions: [ ]
}
```

POST

Creates or updates an account. Note that the client credentials authentication mechanism must be used for account creation. User account access tokens are not authorized to create new accounts. In the response to the creation of a new response, an OAuth2 authorization code will be provided. The calling system can use that authorization code to retrieve an access token for the new account without required the user to explicitly grant access (creating the account assumes access has been granted).

Query Paths

- /account/
- /account/{account-id}

Query Parameters

Pa- rame- ter	Description
account- id	The Carvoyant identifier of the account. If the account-id is not specified, a new account will be created. If it is specified, then any account fields specified in the request will be updated. Unspecified fields will remain unchanged.

Sample JSON Response:

```
{
  account: {
    id: 87
    firstName: "Speed"
    lastName: "Racer"
    username: null
    dateCreated: "20140505T173906+0000"
    email: "matt@carvoyant.com"
    zipcode: "33635"
    phone: null
    timeZone: null
    preferredContact: "EMAIL"
    accessToken: {
      code: "2f2w4ae6mmbvrkd94feen2gy"
    }
  }
}
```

```
totalRecords: null
actions: [ ]
}
```

DELETE

Deletes the specified account.

Warning: This operation is permanent! All data and configuration for the account, including all of its vehicles will be deleted and cannot be restored. Please ensure that the Carvoyant account owner confirms this operation before making the API call.

Query Paths

- /account/{account-id}

Query Parameters

Parameter	Description
account-id	The Carvoyant identifier of the account.

Sample JSON Response:

```
{
  "result": "OK",
  "totalRecords": 1,
  "actions": [ ]
}
```

Vehicle

The Vehicle object represents a unique vehicle within the Carvoyant system. All data with a drivers car/truck is associated with this object.

Properties

Property	Type	Description	Required for Creation
name	String	The text representation of the vehicle type. For instance, “2010 Jeep Wrangler”. If the vehicle type is not known, the value of this property will be “Unidentified Vehicle”	Unsupported
vehicleId	Integer	The internal system identifier for this vehicle.	Unsupported
deviceId	String	The serial number for the Carvoyant device installed in this vehicle.	Optional
vin	String	The Vehicle Identifier Number. Will be null if the vehicle is currently unidentified.	Optional
label	String	A user specified short description for the vehicle.	Optional
autoAssignDevice	Boolean	Indicates whether the system can automatically reassign devices to or from this vehicle. Defaults to true.	Optional
mileage	Integer	The current odometer reading of the vehicle.	Optional
lastWaypoint	Waypoint	The last known geographic location of the vehicle.	Unsupported
running	Boolean	Whether the vehicle is currently running.	Unsupported
lastRunning-Timestamp	Date-Time	The time that the vehicle was last running.	Unsupported
year	String	The Year the vehicle was made.	Unsupported
make	String	The make of the vehicle.	Unsupported
model	String	The model of the vehicle.	Unsupported

Supported Verbs

- GET
- POST
- DELETE

GET

Returns one or more vehicles. By default, the first 50 results are returned.

Query Paths

- /vehicle/
- /vehicle/{vehicle-id}

Query Parameters

Pa- rame- ter	Description
vehicle- id	The Carvoyant identifier of the vehicle. This could be the device serial number in the car (for example, C201200001) or it could be the internal id returned from a previous lookup. If the vehicle-id is not specified, then all vehicles available will be returned.

Call Options

Sortable	No
Pageable	No

Sample JSON Response:

```
{
  "vehicle": {
    "name": "1999 Jeep Wrangler",
    "vehicleId": 3,
    "deviceId": "C201200001",
    "vin": "12345678912345678",
    "label": "Custom dune buggy",
    "mileage": 159774,
    "lastWaypoint": {
      "timestamp": "20140116T152440+0000",
      "latitude": 28.088505,
      "longitude": -82.578467
    },
    "running": false,
    "lastRunningTimestamp": "20140116T151952+0000",
    "year": "1999",
    "make": "Jeep",
    "model": "Wrangler"
  },
  "totalRecords": null,
  "actions": []
}
```

POST

Creates or updates a vehicle.

Query Paths

- /vehicle/
- /vehicle/{vehicle-id}

Query Parameters

Pa- rame- ter	Description
vehicle- id	The Carvoyant identifier of the vehicle. This could be the device serial number in the car (for example, C201200001) or it could be the internal id returned from a previous lookup. If the vehicle-id is not specified, a new vehicle will be created. If it is specified, then any vehicle fields specified in the request will be updated. Unspecified fields will remain unchanged.

DELETE

Deletes the specified vehicle.

Warning: This operation is permanent! All data and configuration for the vehicle will be deleted and cannot be restored. Please ensure that the Carvoyant account owner confirms this operation before making the API call.

Query Paths

- /vehicle/{vehicle-id}

Query Parameters

Parameter	Description
vehicle-id	The Carvoyant identifier of the vehicle. This could be the device serial number in the car (for example, C201200001) or it could be the internal id returned from a previous lookup.

Sample JSON Response:

```
{
  "result": "OK",
  "totalRecords": 1,
  "actions": []
}
```

Trip

This resource represents a Trip within the Carvoyant system. A Trip is defined as the activity of a [Vehicle](#) between ignition on and ignition off.

Properties

Property	Type	Description
id	Integer	The internal system identifier for this trip.
startTime	DateTime	The start time of the trip.
endTime	DateTime	The end time of the trip if the trip is not in progress.
mileage	Float	The distance traveled during the trip.
startWaypoint	Waypoint	The starting location of the trip.
endWaypoint	Waypoint	The ending location of the trip if the trip is in not progress.
data	DataSet Array	An array of data sets collected during the trip.

Supported Verbs

- GET

GET

Returns one or more trips for the specified vehicle. By default, the first 50 results are returned.

Query Paths

- /vehicle/{vehicle-id}/trip/?includeData={true|false}&startTime={startTime}&endTime={endTime}
- /vehicle/{vehicle-id}/trip/{trip-id}

Query Parameters

Parameter	Description
vehicle-id	The Carvoyant identifier of the vehicle. This could be the device serial number in the car (for example, C201200001) or it could be the internal id returned from a previous lookup. If the vehicle-id is not specified, then all vehicles available will be returned.
include-Data	If true, then the results will include all of the DataSets collected during that trip. If false, the trip object will not contain the datum property. Only applies if no trip-id is specified.
start-Time	Used for filtering the results. Only trips that started after this time are returned. See DateTime for the format details. Only applies if no trip-id is specified.
end-Time	Used for filtering the results. Only trips that started before this time are returned. See DateTime for the format details. Only applies if no trip-id is specified.
trip-id	The Carvoyant identifier for the trip.

Call Options

Sortable	Yes (by startTime)
Pageable	Yes (when no trip-id is specified. Individual trip requests are not paginated)

Sample JSON Response (trip list):

```
{
  "trip": [
    {
      "id": 198260,
      "startTime": "20140116T122340+0000",
      "endTime": "20140116T124904+0000",
      "mileage": 0.5,
      "startWaypoint": {
        "timestamp": "20140116T122340+0000",
        "latitude": 28.036539,
        "longitude": -82.593698
      },
      "endWaypoint": {
        "timestamp": "20140116T124904+0000",
        "latitude": 28.036482,
        "longitude": -82.593697
      }
    },
    {
      "id": 197923,
      "startTime": "20140115T230536+0000",
      "endTime": "20140115T231845+0000",
      "mileage": 5.9,
      "startWaypoint": {
        "timestamp": "20140115T230536+0000",
        "latitude": 28.088446,
        "longitude": -82.578471
      },
      "endWaypoint": {
        "timestamp": "20140115T231845+0000",
        "latitude": 28.036513,
        "longitude": -82.593683
      }
    }
  ],
  "totalRecords": 850,
  "actions": [
```

```

    {
      "name": "next",
      "uri": "https://api.carvoyant.com/v1/api/vehicle/C201200001/trip/?includeData=false&sort",
      "methods": null,
      "inputs": null
    },
    {
      "name": "previous",
      "uri": "https://api.carvoyant.com/v1/api/vehicle/C201200001/trip/?includeData=false&sort",
      "methods": null,
      "inputs": null
    }
  ]
}

```

Sample JSON Response (single trip):

```

{
  "trip": {
    "id": 198260,
    "startTime": "20140116T122340+0000",
    "endTime": "20140116T124904+0000",
    "mileage": 0.5,
    "startWaypoint": {
      "timestamp": "20140116T122340+0000",
      "latitude": 28.036539,
      "longitude": -82.593698
    },
    "endWaypoint": {
      "timestamp": "20140116T124904+0000",
      "latitude": 28.036482,
      "longitude": -82.593697
    },
    "data": [
      {
        "id": 2080099,
        "timestamp": "20140116T124904+0000",
        "datum": [
          {
            "id": 8713037,
            "timestamp": "20140116T124904+0000",
            "key": "GEN_TRIP_MILEAGE",
            "value": "0.5",
            "translatedValue": "1 miles"
          },
          {
            "id": 8713036,
            "timestamp": "20140116T124904+0000",
            "key": "GEN_ENGINE_COOLANT_TEMP",
            "value": "81.6",
            "translatedValue": "204.5 deg F"
          },
          {
            "id": 8713035,
            "timestamp": "20140116T124904+0000",
            "key": "GEN_VOLTAGE",
            "value": "13.9",
            "translatedValue": "13.9V"
          }
        ]
      }
    ]
  }
}

```

```

    {
      "id": 8713034,
      "timestamp": "20140116T124904+0000",
      "key": "GEN_SPEED",
      "value": "17.0",
      "translatedValue": "17.0 mph"
    },
    {
      "id": 8713033,
      "timestamp": "20140116T124904+0000",
      "key": "GEN_HEADING",
      "value": "148",
      "translatedValue": "148 deg"
    },
    {
      "id": 8713032,
      "timestamp": "20140116T124904+0000",
      "key": "GEN_WAYPOINT",
      "value": "28.036482,-82.593697",
      "translatedValue": "28.036482,-82.593697"
    }
  ]
},
{
  "id": 2080093,
  "timestamp": "20140116T124839+0000",
  "datum": [
    {
      "id": 8713013,
      "timestamp": "20140116T124839+0000",
      "key": "GEN_ENGINE_COOLANT_TEMP",
      "value": "82.8",
      "translatedValue": "206.6 deg F"
    },
    {
      "id": 8713012,
      "timestamp": "20140116T124839+0000",
      "key": "GEN_VOLTAGE",
      "value": "13.9",
      "translatedValue": "13.9V"
    },
    {
      "id": 8713011,
      "timestamp": "20140116T124839+0000",
      "key": "GEN_SPEED",
      "value": "27.6",
      "translatedValue": "27.6 mph"
    },
    {
      "id": 8713010,
      "timestamp": "20140116T124839+0000",
      "key": "GEN_HEADING",
      "value": "81",
      "translatedValue": "81 deg"
    },
    {
      "id": 8713009,
      "timestamp": "20140116T124839+0000",

```

```

        "key": "GEN_WAYPOINT",
        "value": "28.037042,-82.594338",
        "translatedValue": "28.037042,-82.594338"
      }
    ]
  },
  {
    "id": 2080083,
    "timestamp": "20140116T124739+0000",
    "datum": [
      {
        "id": 8712961,
        "timestamp": "20140116T124739+0000",
        "key": "GEN_ENGINE_COOLANT_TEMP",
        "value": "80.0",
        "translatedValue": "201.6 deg F"
      },
      {
        "id": 8712960,
        "timestamp": "20140116T124739+0000",
        "key": "GEN_VOLTAGE",
        "value": "13.9",
        "translatedValue": "13.9V"
      },
      {
        "id": 8712959,
        "timestamp": "20140116T124739+0000",
        "key": "GEN_HEADING",
        "value": "333",
        "translatedValue": "333 deg"
      },
      {
        "id": 8712958,
        "timestamp": "20140116T124739+0000",
        "key": "GEN_WAYPOINT",
        "value": "28.037880,-82.596209",
        "translatedValue": "28.037880,-82.596209"
      }
    ]
  },
  {
    "id": 2080073,
    "timestamp": "20140116T124639+0000",
    "datum": [
      {
        "id": 8712912,
        "timestamp": "20140116T124639+0000",
        "key": "GEN_ENGINE_COOLANT_TEMP",
        "value": "78.9",
        "translatedValue": "199.6 deg F"
      },
      {
        "id": 8712911,
        "timestamp": "20140116T124639+0000",
        "key": "GEN_VOLTAGE",
        "value": "13.9",
        "translatedValue": "13.9V"
      }
    ],
  },

```

```
{
  "id": 8712910,
  "timestamp": "20140116T124639+0000",
  "key": "GEN_HEADING",
  "value": "333",
  "translatedValue": "333 deg"
},
{
  "id": 8712909,
  "timestamp": "20140116T124639+0000",
  "key": "GEN_WAYPOINT",
  "value": "28.037871,-82.596204",
  "translatedValue": "28.037871,-82.596204"
}
]
},
{
  "id": 2080060,
  "timestamp": "20140116T124539+0000",
  "datum": [
    {
      "id": 8712852,
      "timestamp": "20140116T124539+0000",
      "key": "GEN_ENGINE_COOLANT_TEMP",
      "value": "78.9",
      "translatedValue": "199.6 deg F"
    },
    {
      "id": 8712851,
      "timestamp": "20140116T124539+0000",
      "key": "GEN_VOLTAGE",
      "value": "13.9",
      "translatedValue": "13.9V"
    },
    {
      "id": 8712850,
      "timestamp": "20140116T124539+0000",
      "key": "GEN_SPEED",
      "value": "1.4",
      "translatedValue": "1.4 mph"
    },
    {
      "id": 8712849,
      "timestamp": "20140116T124539+0000",
      "key": "GEN_HEADING",
      "value": "333",
      "translatedValue": "333 deg"
    },
    {
      "id": 8712848,
      "timestamp": "20140116T124539+0000",
      "key": "GEN_WAYPOINT",
      "value": "28.037944,-82.596230",
      "translatedValue": "28.037944,-82.596230"
    }
  ]
},
{
```

```

    "id": 2080053,
    "timestamp": "20140116T124439+0000",
    "datum": [
      {
        "id": 8712823,
        "timestamp": "20140116T124439+0000",
        "key": "GEN_ENGINE_COOLANT_TEMP",
        "value": "78.9",
        "translatedValue": "199.6 deg F"
      },
      {
        "id": 8712822,
        "timestamp": "20140116T124439+0000",
        "key": "GEN_VOLTAGE",
        "value": "13.9",
        "translatedValue": "13.9V"
      },
      {
        "id": 8712821,
        "timestamp": "20140116T124439+0000",
        "key": "GEN_SPEED",
        "value": "2.8",
        "translatedValue": "2.8 mph"
      },
      {
        "id": 8712820,
        "timestamp": "20140116T124439+0000",
        "key": "GEN_HEADING",
        "value": "333",
        "translatedValue": "333 deg"
      },
      {
        "id": 8712819,
        "timestamp": "20140116T124439+0000",
        "key": "GEN_WAYPOINT",
        "value": "28.037912,-82.596222",
        "translatedValue": "28.037912,-82.596222"
      }
    ]
  },
  {
    "id": 2080043,
    "timestamp": "20140116T124339+0000",
    "datum": [
      {
        "id": 8712782,
        "timestamp": "20140116T124339+0000",
        "key": "GEN_ENGINE_COOLANT_TEMP",
        "value": "78.9",
        "translatedValue": "199.6 deg F"
      },
      {
        "id": 8712781,
        "timestamp": "20140116T124339+0000",
        "key": "GEN_VOLTAGE",
        "value": "13.9",
        "translatedValue": "13.9V"
      }
    ],
  }

```

```

    {
      "id": 8712780,
      "timestamp": "20140116T124339+0000",
      "key": "GEN_SPEED",
      "value": "1.1",
      "translatedValue": "1.1 mph"
    },
    {
      "id": 8712779,
      "timestamp": "20140116T124339+0000",
      "key": "GEN_HEADING",
      "value": "133",
      "translatedValue": "133 deg"
    },
    {
      "id": 8712778,
      "timestamp": "20140116T124339+0000",
      "key": "GEN_WAYPOINT",
      "value": "28.037902,-82.596198",
      "translatedValue": "28.037902,-82.596198"
    }
  ]
},
{
  "id": 2080036,
  "timestamp": "20140116T124239+0000",
  "datum": [
    {
      "id": 8712744,
      "timestamp": "20140116T124239+0000",
      "key": "GEN_ENGINE_COOLANT_TEMP",
      "value": "80.0",
      "translatedValue": "201.6 deg F"
    },
    {
      "id": 8712743,
      "timestamp": "20140116T124239+0000",
      "key": "GEN_VOLTAGE",
      "value": "13.9",
      "translatedValue": "13.9V"
    },
    {
      "id": 8712742,
      "timestamp": "20140116T124239+0000",
      "key": "GEN_HEADING",
      "value": "165",
      "translatedValue": "165 deg"
    },
    {
      "id": 8712741,
      "timestamp": "20140116T124239+0000",
      "key": "GEN_WAYPOINT",
      "value": "28.037869,-82.596201",
      "translatedValue": "28.037869,-82.596201"
    }
  ]
},
{

```

```

    "id": 2080028,
    "timestamp": "20140116T124139+0000",
    "datum": [
      {
        "id": 8712705,
        "timestamp": "20140116T124139+0000",
        "key": "GEN_ENGINE_COOLANT_TEMP",
        "value": "80.5",
        "translatedValue": "202.5 deg F"
      },
      {
        "id": 8712704,
        "timestamp": "20140116T124139+0000",
        "key": "GEN_VOLTAGE",
        "value": "13.9",
        "translatedValue": "13.9V"
      },
      {
        "id": 8712703,
        "timestamp": "20140116T124139+0000",
        "key": "GEN_SPEED",
        "value": "4.7",
        "translatedValue": "4.7 mph"
      },
      {
        "id": 8712702,
        "timestamp": "20140116T124139+0000",
        "key": "GEN_HEADING",
        "value": "165",
        "translatedValue": "165 deg"
      },
      {
        "id": 8712701,
        "timestamp": "20140116T124139+0000",
        "key": "GEN_WAYPOINT",
        "value": "28.037881,-82.596196",
        "translatedValue": "28.037881,-82.596196"
      }
    ]
  },
  {
    "id": 2080016,
    "timestamp": "20140116T124039+0000",
    "datum": [
      {
        "id": 8712653,
        "timestamp": "20140116T124039+0000",
        "key": "GEN_ENGINE_COOLANT_TEMP",
        "value": "77.8",
        "translatedValue": "197.6 deg F"
      },
      {
        "id": 8712652,
        "timestamp": "20140116T124039+0000",
        "key": "GEN_VOLTAGE",
        "value": "14.0",
        "translatedValue": "14.0V"
      }
    ],
  },

```

```
{
  {
    "id": 8712651,
    "timestamp": "20140116T124039+0000",
    "key": "GEN_SPEED",
    "value": "1.5",
    "translatedValue": "1.5 mph"
  },
  {
    "id": 8712650,
    "timestamp": "20140116T124039+0000",
    "key": "GEN_HEADING",
    "value": "351",
    "translatedValue": "351 deg"
  },
  {
    "id": 8712649,
    "timestamp": "20140116T124039+0000",
    "key": "GEN_WAYPOINT",
    "value": "28.037978,-82.596189",
    "translatedValue": "28.037978,-82.596189"
  }
]
},
{
  "id": 2080009,
  "timestamp": "20140116T123939+0000",
  "datum": [
    {
      "id": 8712622,
      "timestamp": "20140116T123939+0000",
      "key": "GEN_ENGINE_COOLANT_TEMP",
      "value": "78.9",
      "translatedValue": "199.6 deg F"
    },
    {
      "id": 8712621,
      "timestamp": "20140116T123939+0000",
      "key": "GEN_VOLTAGE",
      "value": "14.0",
      "translatedValue": "14.0V"
    },
    {
      "id": 8712620,
      "timestamp": "20140116T123939+0000",
      "key": "GEN_HEADING",
      "value": "329",
      "translatedValue": "329 deg"
    },
    {
      "id": 8712619,
      "timestamp": "20140116T123939+0000",
      "key": "GEN_WAYPOINT",
      "value": "28.037968,-82.596198",
      "translatedValue": "28.037968,-82.596198"
    }
  ]
},
{
```

```

    "id": 2080000,
    "timestamp": "20140116T123839+0000",
    "datum": [
      {
        "id": 8712583,
        "timestamp": "20140116T123839+0000",
        "key": "GEN_ENGINE_COOLANT_TEMP",
        "value": "83.9",
        "translatedValue": "208.6 deg F"
      },
      {
        "id": 8712582,
        "timestamp": "20140116T123839+0000",
        "key": "GEN_VOLTAGE",
        "value": "14.0",
        "translatedValue": "14.0V"
      },
      {
        "id": 8712581,
        "timestamp": "20140116T123839+0000",
        "key": "GEN_HEADING",
        "value": "329",
        "translatedValue": "329 deg"
      },
      {
        "id": 8712580,
        "timestamp": "20140116T123839+0000",
        "key": "GEN_WAYPOINT",
        "value": "28.037968,-82.596202",
        "translatedValue": "28.037968,-82.596202"
      }
    ]
  },
  {
    "id": 2079993,
    "timestamp": "20140116T123739+0000",
    "datum": [
      {
        "id": 8712547,
        "timestamp": "20140116T123739+0000",
        "key": "GEN_ENGINE_COOLANT_TEMP",
        "value": "80.5",
        "translatedValue": "202.5 deg F"
      },
      {
        "id": 8712546,
        "timestamp": "20140116T123739+0000",
        "key": "GEN_VOLTAGE",
        "value": "14.0",
        "translatedValue": "14.0V"
      },
      {
        "id": 8712545,
        "timestamp": "20140116T123739+0000",
        "key": "GEN_HEADING",
        "value": "329",
        "translatedValue": "329 deg"
      }
    ],
  }

```

```
{
  "id": 8712544,
  "timestamp": "20140116T123739+0000",
  "key": "GEN_WAYPOINT",
  "value": "28.037976,-82.596200",
  "translatedValue": "28.037976,-82.596200"
}
],
{
  "id": 2079986,
  "timestamp": "20140116T123639+0000",
  "datum": [
    {
      "id": 8712514,
      "timestamp": "20140116T123639+0000",
      "key": "GEN_ENGINE_COOLANT_TEMP",
      "value": "78.9",
      "translatedValue": "199.6 deg F"
    },
    {
      "id": 8712513,
      "timestamp": "20140116T123639+0000",
      "key": "GEN_VOLTAGE",
      "value": "13.9",
      "translatedValue": "13.9V"
    },
    {
      "id": 8712512,
      "timestamp": "20140116T123639+0000",
      "key": "GEN_HEADING",
      "value": "329",
      "translatedValue": "329 deg"
    },
    {
      "id": 8712511,
      "timestamp": "20140116T123639+0000",
      "key": "GEN_WAYPOINT",
      "value": "28.037955,-82.596192",
      "translatedValue": "28.037955,-82.596192"
    }
  ]
},
{
  "id": 2079977,
  "timestamp": "20140116T123539+0000",
  "datum": [
    {
      "id": 8712476,
      "timestamp": "20140116T123539+0000",
      "key": "GEN_ENGINE_COOLANT_TEMP",
      "value": "76.6",
      "translatedValue": "195.5 deg F"
    },
    {
      "id": 8712475,
      "timestamp": "20140116T123539+0000",
      "key": "GEN_VOLTAGE",
```

```

        "value": "14.0",
        "translatedValue": "14.0V"
    },
    {
        "id": 8712474,
        "timestamp": "20140116T123539+0000",
        "key": "GEN_HEADING",
        "value": "329",
        "translatedValue": "329 deg"
    },
    {
        "id": 8712473,
        "timestamp": "20140116T123539+0000",
        "key": "GEN_WAYPOINT",
        "value": "28.037959,-82.596177",
        "translatedValue": "28.037959,-82.596177"
    }
]
},
{
    "id": 2079973,
    "timestamp": "20140116T123439+0000",
    "datum": [
        {
            "id": 8712463,
            "timestamp": "20140116T123439+0000",
            "key": "GEN_ENGINE_COOLANT_TEMP",
            "value": "75.0",
            "translatedValue": "192.6 deg F"
        },
        {
            "id": 8712462,
            "timestamp": "20140116T123439+0000",
            "key": "GEN_VOLTAGE",
            "value": "14.0",
            "translatedValue": "14.0V"
        },
        {
            "id": 8712461,
            "timestamp": "20140116T123439+0000",
            "key": "GEN_HEADING",
            "value": "329",
            "translatedValue": "329 deg"
        },
        {
            "id": 8712460,
            "timestamp": "20140116T123439+0000",
            "key": "GEN_WAYPOINT",
            "value": "28.037962,-82.596198",
            "translatedValue": "28.037962,-82.596198"
        }
    ]
},
{
    "id": 2079967,
    "timestamp": "20140116T123339+0000",
    "datum": [

```

```

        "id": 8712437,
        "timestamp": "20140116T123339+0000",
        "key": "GEN_ENGINE_COOLANT_TEMP",
        "value": "72.8",
        "translatedValue": "188.6 deg F"
    },
    {
        "id": 8712436,
        "timestamp": "20140116T123339+0000",
        "key": "GEN_VOLTAGE",
        "value": "13.9",
        "translatedValue": "13.9V"
    },
    {
        "id": 8712435,
        "timestamp": "20140116T123339+0000",
        "key": "GEN_HEADING",
        "value": "329",
        "translatedValue": "329 deg"
    },
    {
        "id": 8712434,
        "timestamp": "20140116T123339+0000",
        "key": "GEN_WAYPOINT",
        "value": "28.037976,-82.596202",
        "translatedValue": "28.037976,-82.596202"
    }
]
},
{
    "id": 2079957,
    "timestamp": "20140116T123239+0000",
    "datum": [
        {
            "id": 8712402,
            "timestamp": "20140116T123239+0000",
            "key": "GEN_ENGINE_COOLANT_TEMP",
            "value": "70.0",
            "translatedValue": "183.6 deg F"
        },
        {
            "id": 8712401,
            "timestamp": "20140116T123239+0000",
            "key": "GEN_VOLTAGE",
            "value": "14.0",
            "translatedValue": "14.0V"
        },
        {
            "id": 8712400,
            "timestamp": "20140116T123239+0000",
            "key": "GEN_HEADING",
            "value": "329",
            "translatedValue": "329 deg"
        },
        {
            "id": 8712399,
            "timestamp": "20140116T123239+0000",
            "key": "GEN_WAYPOINT",

```

```

        "value": "28.037980,-82.596212",
        "translatedValue": "28.037980,-82.596212"
    }
]
},
{
    "id": 2079949,
    "timestamp": "20140116T123139+0000",
    "datum": [
        {
            "id": 8712366,
            "timestamp": "20140116T123139+0000",
            "key": "GEN_ENGINE_COOLANT_TEMP",
            "value": "67.8",
            "translatedValue": "179.6 deg F"
        },
        {
            "id": 8712365,
            "timestamp": "20140116T123139+0000",
            "key": "GEN_VOLTAGE",
            "value": "14.0",
            "translatedValue": "14.0V"
        },
        {
            "id": 8712364,
            "timestamp": "20140116T123139+0000",
            "key": "GEN_SPEED",
            "value": "1.1",
            "translatedValue": "1.1 mph"
        },
        {
            "id": 8712363,
            "timestamp": "20140116T123139+0000",
            "key": "GEN_HEADING",
            "value": "329",
            "translatedValue": "329 deg"
        },
        {
            "id": 8712362,
            "timestamp": "20140116T123139+0000",
            "key": "GEN_WAYPOINT",
            "value": "28.037986,-82.596210",
            "translatedValue": "28.037986,-82.596210"
        }
    ]
},
{
    "id": 2079941,
    "timestamp": "20140116T123039+0000",
    "datum": [
        {
            "id": 8712329,
            "timestamp": "20140116T123039+0000",
            "key": "GEN_ENGINE_COOLANT_TEMP",
            "value": "65.0",
            "translatedValue": "174.6 deg F"
        },
        {

```

```

        "id": 8712328,
        "timestamp": "20140116T123039+0000",
        "key": "GEN_VOLTAGE",
        "value": "13.9",
        "translatedValue": "13.9V"
    },
    {
        "id": 8712327,
        "timestamp": "20140116T123039+0000",
        "key": "GEN_HEADING",
        "value": "351",
        "translatedValue": "351 deg"
    },
    {
        "id": 8712326,
        "timestamp": "20140116T123039+0000",
        "key": "GEN_WAYPOINT",
        "value": "28.037966,-82.596160",
        "translatedValue": "28.037966,-82.596160"
    }
]
},
{
    "id": 2079933,
    "timestamp": "20140116T122939+0000",
    "datum": [
        {
            "id": 8712303,
            "timestamp": "20140116T122939+0000",
            "key": "GEN_ENGINE_COOLANT_TEMP",
            "value": "60.5",
            "translatedValue": "166.5 deg F"
        },
        {
            "id": 8712302,
            "timestamp": "20140116T122939+0000",
            "key": "GEN_VOLTAGE",
            "value": "14.0",
            "translatedValue": "14.0V"
        },
        {
            "id": 8712301,
            "timestamp": "20140116T122939+0000",
            "key": "GEN_HEADING",
            "value": "351",
            "translatedValue": "351 deg"
        },
        {
            "id": 8712300,
            "timestamp": "20140116T122939+0000",
            "key": "GEN_WAYPOINT",
            "value": "28.037975,-82.596183",
            "translatedValue": "28.037975,-82.596183"
        }
    ]
},
{
    "id": 2079926,

```

```

"timestamp": "20140116T122839+0000",
"datum": [
  {
    "id": 8712267,
    "timestamp": "20140116T122839+0000",
    "key": "GEN_ENGINE_COOLANT_TEMP",
    "value": "55.5",
    "translatedValue": "157.5 deg F"
  },
  {
    "id": 8712266,
    "timestamp": "20140116T122839+0000",
    "key": "GEN_VOLTAGE",
    "value": "14.0",
    "translatedValue": "14.0V"
  },
  {
    "id": 8712265,
    "timestamp": "20140116T122839+0000",
    "key": "GEN_SPEED",
    "value": "3.2",
    "translatedValue": "3.2 mph"
  },
  {
    "id": 8712264,
    "timestamp": "20140116T122839+0000",
    "key": "GEN_HEADING",
    "value": "351",
    "translatedValue": "351 deg"
  },
  {
    "id": 8712263,
    "timestamp": "20140116T122839+0000",
    "key": "GEN_WAYPOINT",
    "value": "28.037986,-82.596209",
    "translatedValue": "28.037986,-82.596209"
  }
]
},
{
  "id": 2079920,
  "timestamp": "20140116T122739+0000",
  "datum": [
    {
      "id": 8712243,
      "timestamp": "20140116T122739+0000",
      "key": "GEN_ENGINE_COOLANT_TEMP",
      "value": "50.0",
      "translatedValue": "147.6 deg F"
    },
    {
      "id": 8712242,
      "timestamp": "20140116T122739+0000",
      "key": "GEN_VOLTAGE",
      "value": "14.0",
      "translatedValue": "14.0V"
    }
  ],
  {

```

```

        "id": 8712241,
        "timestamp": "20140116T122739+0000",
        "key": "GEN_HEADING",
        "value": "166",
        "translatedValue": "166 deg"
    },
    {
        "id": 8712240,
        "timestamp": "20140116T122739+0000",
        "key": "GEN_WAYPOINT",
        "value": "28.037970,-82.596211",
        "translatedValue": "28.037970,-82.596211"
    }
]
},
{
    "id": 2079914,
    "timestamp": "20140116T122639+0000",
    "datum": [
        {
            "id": 8712212,
            "timestamp": "20140116T122639+0000",
            "key": "GEN_ENGINE_COOLANT_TEMP",
            "value": "43.9",
            "translatedValue": "136.6 deg F"
        },
        {
            "id": 8712211,
            "timestamp": "20140116T122639+0000",
            "key": "GEN_VOLTAGE",
            "value": "14.0",
            "translatedValue": "14.0V"
        },
        {
            "id": 8712210,
            "timestamp": "20140116T122639+0000",
            "key": "GEN_HEADING",
            "value": "166",
            "translatedValue": "166 deg"
        },
        {
            "id": 8712209,
            "timestamp": "20140116T122639+0000",
            "key": "GEN_WAYPOINT",
            "value": "28.037971,-82.596211",
            "translatedValue": "28.037971,-82.596211"
        }
    ]
},
{
    "id": 2079913,
    "timestamp": "20140116T122539+0000",
    "datum": [
        {
            "id": 8712208,
            "timestamp": "20140116T122539+0000",
            "key": "GEN_ENGINE_COOLANT_TEMP",
            "value": "35.5",

```

```

        "translatedValue": "121.5 deg F"
    },
    {
        "id": 8712207,
        "timestamp": "20140116T122539+0000",
        "key": "GEN_VOLTAGE",
        "value": "14.0",
        "translatedValue": "14.0V"
    },
    {
        "id": 8712206,
        "timestamp": "20140116T122539+0000",
        "key": "GEN_SPEED",
        "value": "22.5",
        "translatedValue": "22.5 mph"
    },
    {
        "id": 8712205,
        "timestamp": "20140116T122539+0000",
        "key": "GEN_HEADING",
        "value": "166",
        "translatedValue": "166 deg"
    },
    {
        "id": 8712204,
        "timestamp": "20140116T122539+0000",
        "key": "GEN_WAYPOINT",
        "value": "28.037975,-82.596213",
        "translatedValue": "28.037975,-82.596213"
    }
]
},
{
    "id": 2079910,
    "timestamp": "20140116T122439+0000",
    "datum": [
        {
            "id": 8712191,
            "timestamp": "20140116T122439+0000",
            "key": "GEN_ENGINE_COOLANT_TEMP",
            "value": "22.8",
            "translatedValue": "98.6 deg F"
        },
        {
            "id": 8712190,
            "timestamp": "20140116T122439+0000",
            "key": "GEN_VOLTAGE",
            "value": "14.1",
            "translatedValue": "14.1V"
        },
        {
            "id": 8712189,
            "timestamp": "20140116T122439+0000",
            "key": "GEN_SPEED",
            "value": "23.5",
            "translatedValue": "23.5 mph"
        }
    ]
}

```

```
        "id": 8712188,  
        "timestamp": "20140116T122439+0000",  
        "key": "GEN_HEADING",  
        "value": "330",  
        "translatedValue": "330 deg"  
    },  
    {  
        "id": 8712187,  
        "timestamp": "20140116T122439+0000",  
        "key": "GEN_WAYPOINT",  
        "value": "28.037493,-82.595962",  
        "translatedValue": "28.037493,-82.595962"  
    }  
]  
},  
{  
    "id": 2079908,  
    "timestamp": "20140116T122340+0000",  
    "datum": [  
        {  
            "id": 8712185,  
            "timestamp": "20140116T122340+0000",  
            "key": "GEN_ENGINE_COOLANT_TEMP",  
            "value": "83.9",  
            "translatedValue": "208.6 deg F"  
        },  
        {  
            "id": 8712184,  
            "timestamp": "20140116T122340+0000",  
            "key": "GEN_VOLTAGE",  
            "value": "14.0",  
            "translatedValue": "14.0V"  
        },  
        {  
            "id": 8712183,  
            "timestamp": "20140116T122340+0000",  
            "key": "GEN_HEADING",  
            "value": "326",  
            "translatedValue": "326 deg"  
        },  
        {  
            "id": 8712182,  
            "timestamp": "20140116T122340+0000",  
            "key": "GEN_WAYPOINT",  
            "value": "28.036539,-82.593698",  
            "translatedValue": "28.036539,-82.593698"  
        }  
    ]  
}  
]  
},  
"totalRecords": 27,  
"actions": []  
}
```

DataSet

A DataSet is a collection of [DataPoint](#) s taken at the same time. For example, a DataSet may contain a speed value, a battery voltage value and a GPS location. It should be assumed that the DataPoints are related and were collected at the same time. This allows you to correlate different DataPoints with each other.

Properties

Property	Type	Description	Required for Creation
id	Integer	The internal system identifier for this data set.	Unsupported
vin	String	The VIN of the vehicle the data is from	Optional
deviceId	String	The deviceId of the device collecting the data	Required
timestamp	DateTime	The time that the DataSet was created.	Required
datum	DataPoint	The DataPoint s contained within this DataSet.	Required
ignition-Status	String Enumeration ON OFF RUNNING	The state of the vehicle when this DataSet was collected.	Required

Supported Verbs

- GET
- POST

GET

Returns data sets for the specified vehicle. By default, the first 50 results are returned.

Query Paths

- /vehicle/{vehicle-id}/dataSet/
- /vehicle/{vehicle-id}/dataSet/?startTime={startTime}&endTime={endTime}

Query Parameters

Parameter	Description
start-Time	Used for filtering the results. Only data sets that started after this time are returned. See DateTime for the format details.
endTime	Used for filtering the results. Only data sets that started before this time are returned. See DateTime for the format details.

Call Options

Sortable	Yes (by timestamp)
Pageable	Yes

Sample JSON Response:

```
{
  "dataSet": [
    {
      "id": 1187922,
      "timestamp": "19991130T000000+0000",
      "datum": [
        {
```

```

        "id": 4562158,
        "timestamp": "19991130T000000+0000",
        "key": "GEN_WAYPOINT",
        "value": "0.000000,0.000000",
        "translatedValue": "0.000000,0.000000"
      },
      {
        "id": 4562160,
        "timestamp": "19991130T000000+0000",
        "key": "GEN_VOLTAGE",
        "value": "13.9",
        "translatedValue": "13.9V"
      },
      {
        "id": 4562163,
        "timestamp": "19991130T000000+0000",
        "key": "GEN_FUELRATE",
        "value": "0.0",
        "translatedValue": "0.0 gph"
      }
    ]
  },
  {
    "id": 1562501,
    "timestamp": "19991130T000000+0000",
    "datum": [
      {
        "id": 6223164,
        "timestamp": "19991130T000000+0000",
        "key": "GEN_WAYPOINT",
        "value": "0.000000,0.000000",
        "translatedValue": "0.000000,0.000000"
      },
      {
        "id": 6223165,
        "timestamp": "19991130T000000+0000",
        "key": "GEN_VOLTAGE",
        "value": "13.9",
        "translatedValue": "13.9V"
      },
      {
        "id": 6223167,
        "timestamp": "19991130T000000+0000",
        "key": "GEN_FUELLEVEL",
        "value": "0",
        "translatedValue": "0 %"
      }
    ]
  }
],
"totalRecords": 23893,
"actions": [
  {
    "name": "next",
    "uri": "https://api.carvoyant.com/v1/api/vehicle/C201200001/dataSet/?searchOffset=2&search",
    "methods": null,
    "inputs": null
  }
]

```

```
]
}
```

POST

Saves the data set to the specified vehicle. Note that in the production environment, this is a restricted call that only certain partners are authorized to use. If you feel you need to make calls to this endpoint, please [contact us](#). In the sandbox environment, this is available for everyone.

Be aware that the system expects that a waypoint will be included with all DataSets that are posted to the system. It will not fail without one, but the user expects location to be available for all recorded data points.

Query Paths

- /vehicle/{vehicle-id}/dataSet/

Sample JSON Request:

```
{
  "timestamp": "20140811T140444+0000",
  "vin": "123456789ABCDEFGH",
  "deviceId": "C20120000X",
  "ignitionStatus": "ON",
  "datum": [
    {
      "timestamp": "20140811T140444+0000",
      "key": "GEN_WAYPOINT",
      "value": "28.027065,-82.588619"
    },
    {
      "timestamp": "20140811T140444+0000",
      "key": "GEN_HEADING",
      "value": 323
    },
    {
      "timestamp": "20140811T140444+0000",
      "key": "GEN_VOLTAGE",
      "value": "13.6"
    }
  ]
}
```

DataPoint

Represents a data point collected from a vehicle.

Properties

Property	Type	Description
id	Integer	The internal system identifier for this data point.
timestamp	Date-Time	The time that the data point was collected.
key	DataKey	The type of data collected.
value	String	The raw value collected from the vehicle.
translated-Value	String	The value collected translated into human understandable form. This should only be used for display purposes.

Supported Verbs

- GET

GET

Returns raw vehicle data for the specified vehicle. By default, the first 50 results are returned.

Query Paths

- /vehicle/{vehicle-id}/data/
- /vehicle/{vehicle-id}/data/?key={key-id}
- /vehicle/{vehicle-id}/data/?mostRecentOnly=true

Query Parameters

Parameter	Description
key	Allows only specific data types to be returned. See DataKey for the allowable key-ids.
mostRecentOnly	If this is present, only the most recently collected data point for each type will be returned.

Call Options

Sortable	Yes (by timestamp)
Pageable	Yes

Sample JSON Response:

```
{
  "data": [
    {
      "id": 8727772,
      "timestamp": "20140116T183654+0000",
      "key": "GEN_WAYPOINT",
      "value": "28.088472,-82.578439",
      "translatedValue": "28.088472,-82.578439"
    },
    {
      "id": 8727773,
      "timestamp": "20140116T183654+0000",
      "key": "GEN_VOLTAGE",
      "value": "12.6",
      "translatedValue": "12.6V"
    }
  ],
  "totalRecords": 65070,
  "actions": [
    {
      "name": "next",
      "uri": "https://api.carvoyant.com/v1/api/vehicle/C201200001/data/?sortOrder=desc&searchO",
      "methods": null,
      "inputs": null
    }
  ]
}
```

EventSubscription

The EventSubscription object represents a unique event subscription within the Carvoyant system. The user may subscribe to an event that will deliver an [EventNotification](#) to the defined postURL when vehicle data has reached the specified criteria. In order to create an EventSubscription, all required fields for the specific event “_type” must be delivered in the body of the HTTP POST request. Required fields, resource URIs, and examples for the supported event types can be found in this documents children.

The Event model that we have implemented is based off of the [Evented API Spec](#). This is a generic specification that helps define the transport of API events between two systems.

See the [EventType](#) page for details on the different events that can be subscribed to.

Note: You will only be able to interact with subscriptions that have been created with your client Id. Specifically, we will look at the access token specified in the request, determine the client Id that was authorized with that access token, and only return subscriptions for that client Id. You do not have access to subscriptions that have been created by other client Ids.

Common Properties

Property	Type	Description	Required for Creation
id	Integer	The internal system identifier for this subscription.	Unsupported
_type	Event-Type	The type of event being subscribed to. (In this case the _type is “LOWBATTERY”)	Unsupported
creation-Timestamp	Date-Time	The time when the subscription was created.	Unsupported
deletion-Timestamp	Date-Time	The time when the subscription was marked for deletion. (This value will only be returned if the event has been marked for deletion)	Unsupported
minimum-Time	Integer	The time in minutes that will determine the minimum interval between event notification creation. If the value is less than the reporting interval of the hardware, the hardware limit will be used.	Required
creator-ClientId	String	The client Id that generated this subscription.	Unsupported
postUrl	String	The URL that will receive HTTPS POST notifications from the newly generated subscription. Note that only HTTPS endpoints are supported. The certificate protecting the post URL must be a valid signed certificate.	Deprecated Use deliveries instead
post-Headers	Map	A map that contains any headers that should be sent to the postUrl when a notification is generated.	Deprecated Use deliveries instead
notification-Period	NotificationPeriod	A string that represents when EventNotifications are sent from the Carvoyant system. All NotificationPeriod types are supported for LowBattery subscriptions.	Required
deliveries	Array of Delivery-Type	An array with the delivery types for this subscription	Required

Automated Updates

We have implemented some logic to automatically update subscriptions based on responses to the notifications that are sent to them. When the Carvoyant system attempts to send a notification to a HTTP_POST delivery option, the

following actions will be taken based on the response code.

- HTTP 301 (Moved Permanently) and HTTP 308 (Permanent Redirect) - The `postUrl` of the delivery will be updated with the value in the `Location` response header. The notification will then be resent to the new `postUrl`.
- HTTP 302 (Found) and HTTP 307 (Temporary Redirect) - The notification will be resent to the url specified in the `Location` response header but the `postUrl` of the delivery option will remain unchanged.
- HTTP 4xx - We assume these are fatal errors and the delivery option will be deleted.
- HTTP 5xx - We assume these are temporary server side errors. The current notification will not be attempted again but the delivery option will remain unchanged.

Supported Verbs

- GET
- POST
- DELETE

GET

Returns one or more event subscriptions. By default, the first 50 results are returned.

Query Paths

- `/account/{account-id}/eventSubscription/{subscription-id}`
- `/account/{account-id}/eventSubscription/{event-type}/{subscription-id}`
- `/vehicle/{vehicle-id}/eventSubscription/{subscription-id}`
- `/vehicle/{vehicle-id}/eventSubscription/{event-type}/{subscription-id}`

Query Parameters

Parameter	Description
<code>account-id</code>	The Carvoyant identifier of the account. This is used for account level subscriptions
<code>vehicle-id</code>	The Carvoyant identifier of the vehicle. This could be the device serial number in the car (for example, C201200001) or it could be the internal id returned from a previous lookup. This is used for vehicle level subscriptions
<code>subscription-id</code>	The Carvoyant identifier of the subscription. If the <code>subscription-id</code> is not specified, then all subscriptions available will be returned.
<code>event-type</code>	Indicates the EventType of subscriptions to be returned.

Call Options

<code>Sortable</code>	Yes (by timestamp)
<code>Pageable</code>	Yes (when no <code>subscription-id</code> is specified. Individual subscription requests are not paginated)

Sample JSON Response:

```
{
  "subscriptions": [{
    "id": 1645,
    "_type": "LOWBATTERY",
```

```

    "_timestamp": "20140911T203312+0000",
    "minimumTime": 0,
    "creatorClientId": "hasa2czfebhsj6XXXXXXXXXX",
    "vehicleId": 123,
    "postUrl": "https://test.carvoyant.com/notify",
    "postHeaders": {
      "Authorization": "Bearer asdfqwerzxcv",
      "X-Sample-Headers": "Some custom value"
    },
    "notificationPeriod": "STATECHANGE"
  }, {
    "id": 1646,
    "_type": "VEHICLECONNECTED",
    "_timestamp": "20140911T203348+0000",
    "minimumTime": 0,
    "creatorClientId": "hasa2czfebhsj6XXXXXXXXXX",
    "vehicleId": 123,
    "postUrl": "https://test.carvoyant.com/notify",
    "postHeaders": {},
    "notificationPeriod": "INITIALSTATE"
  }, {
    "id": 1647,
    "_type": "VEHICLEDISCONNECTED",
    "_timestamp": "20140911T203408+0000",
    "minimumTime": 0,
    "creatorClientId": "hasa2czfebhsj6XXXXXXXXXX",
    "vehicleId": 123,
    "postUrl": "https://test.carvoyant.com/notify",
    "postHeaders": {},
    "notificationPeriod": "INITIALSTATE"
  }
],
"totalRecords": 3
}

```

POST

Creates a subscription. The query parameters listed here are common to all [EventType](#). In order to successfully create a subscription the body of the request must specify all required properties of the particular [EventType](#).

Note: Existing subscriptions cannot be updated. To “change” a subscription, you must delete the old one and create a new one.

Query Paths

- /account/{account-id}/eventSubscription/{event-type}/
- /vehicle/{vehicle-id}/eventSubscription/{event-type}/

Query Parameters

Parameter	Description
account-id	The Carvoyant identifier of the account. This is used for account level subscriptions
vehicle-id	The Carvoyant identifier of the vehicle. This could be the device serial number in the car (for example, C201200001) or it could be the internal id returned from a previous lookup. This is used for vehicle level subscriptions
event-type	Indicates the EventType of subscriptions to be returned.

Sample Request:

```
{
  "minimumTime": 0,
  "postUrl": "https://test.carvoyant.com/notify",
  "postHeaders": {
    "Authorization": "Bearer asdfqwerzxcv",
    "X-Sample-Headers": "Some custom value"
  },
  "notificationPeriod": "CONTINUOUS"
}
```

DELETE

Marks a subscription for deletion. The system will purge the subscription after a set amount of time. These are not immediately deleted because doing so would also delete the history of [EventNotifications](#) for this subscription.

Query Paths

- /account/{account-id}/eventSubscription/{subscription-id}
- /account/{account-id}/eventSubscription/{event-type}/{subscription-id}
- /vehicle/{vehicle-id}/eventSubscription/{subscription-id}
- /vehicle/{vehicle-id}/eventSubscription/{event-type}/{subscription-id}

Query Parameters

Parameter	Description
account-id	The Carvoyant identifier of the account. This is used for account level subscriptions
vehicle-id	The Carvoyant identifier of the vehicle. This could be the device serial number in the car (for example, C201200001) or it could be the internal id returned from a previous lookup. This is used for vehicle level subscriptions
subscription-id	The Carvoyant identifier of the subscription. If the subscription-id is not specified, then all subscriptions available will be returned.
event-type	Indicates the EventType of subscriptions to be returned.

Sample JSON Response:

```
{
  "result": "OK",
  "totalRecords": 1,
  "actions": []
}
```

EventNotification

The EventNotification object corresponds to a unique [EventSubscription](#) within the Carvoyant system. The EventNotification will be delivered to the postURL provided by the [EventSubscription](#) via an HTTP POST message whose JSON body contains the details specific to the notification type. Examples for the supported event types can be found in this documents children.

The Event model that we have implemented is based off of the [Evented API Spec](#). This is a generic specification that helps define the transport of API events between two systems.

See the [EventType](#) page for details on the different events that can be notified.

Note: You will only be able to get notifications that have been created for your client Id. Specifically, we will look at the access token specified in the request, determine the client Id that was authorized with that access token, and only return notifications for [EventSubscription](#) s for that client Id. You do not have access to notifications for subscriptions that have been created by other client Ids.

Common Properties

Property	Type	Description
id	Integer	The internal system identifier for this notification.
accountId	Integer	The internal system identifier for this account.
vehicleId	Integer	The internal system identifier for this vehicle.
dataSetId	Integer	The Id of the DataSet that generated the notification.
tripId	Integer	The Id of the Trip that generated the notification.
subscriptionId	Integer	The internal system identifier for this notification's corresponding EventSubscription .
_domain	String	Serves as a namespace for the event.
_type	Event-Type	The type of event notification being sent.
_name	Event-Type	The type of event notification being sent. Note that this field should be considered deprecated. It is only supported because the Evented API spec has not fully implemented the change.
_timestamp	Date-Time	The time when the event occurred.
eventTimestamp	Date-Time	The time when the event notification was created.
minimumTime	Integer	The time in minutes that will determine the minimum interval between event notification creation. If the value is less than the reporting interval of the hardware, the hardware limit will be used.
statusCode	Integer	The status code that was returned when attempting to POST the EventNotification to the postUrl. When the EventNotification is delivered to the postUrl the value of statusCode will be 0. However, a subsequent GET request on the EventNotification will provide a useful statusCode value derived from the postUrl's response.
notificationPeriod	NotificationPeriod	A string that represents when EventNotifications are sent from the Carvoyant system. Each EventType will define what periods are supported.

Supported Verbs

- GET

GET

Returns one or more event notifications. By default, the first 50 results are returned.

Query Paths

- /account/{account-id}/eventNotification/{notification-id}
- /account/{account-id}/eventNotification/{event-type}/{notification-id}
- /account/{account-id}/eventSubscription/{subscription-id}/eventNotification/{notification-id}
- /account/{account-id}/eventSubscription/{subscription-id}/eventNotification/{event-type}/{notification-id}
- /vehicle/{vehicle-id}/eventNotification/{notification-id}
- /vehicle/{vehicle-id}/eventNotification/{event-type}/{notification-id}
- /vehicle/{vehicle-id}/eventSubscription/{subscription-id}/eventNotification/{notification-id}
- /vehicle/{vehicle-id}/eventSubscription/{subscription-id}/eventNotification/{event-type}/{notification-id}

Query Parameters

Parameter	Description
account-id	The Carvoyant identifier of the account. This is used for account level notification
vehicle-id	The Carvoyant identifier of the vehicle. This could be the device serial number in the car (for example, C201200001) or it could be the internal id returned from a previous lookup. This is used for vehicle level notification
notification-id	The Carvoyant identifier of the notification. If the notification-id is not specified, then all notifications available will be returned.
subscription-id	The Carvoyant identifier of the subscription.
event-type	Indicates the EventType of notifications to be returned.

Call Options

Sortable	Yes (by timestamp)
Pageable	Yes (when no notification-id is specified. Individual notification requests are not paginated)

Sample JSON Response

Note: This response only includes the properties that are common to all [EventType](#) . It is not a complete response. Refer to the [EventType](#) page for the detailed list of what properties are returned for the notification.

```
{
  "notifications": [
    {
      "id": 315931,
      "subscriptionId": 1647,
      "_domain": "carvoyant.com",
      "_type": "VEHICLEDISCONNECTED",
      "_name": "VEHICLEDISCONNECTED",
      "_timestamp": "20140912T010246+0000",
      "minimumTime": 0,
      "httpStatusCode": 200,
      "notificationPeriod": "INITIALSTATE",
    }
  ]
}
```

```

    "dataSetId":4795420,
    "creatorClientId":"hasa2czfebhsj6XXXXXXXXXX",
    "vehicleId":123
  },
  {
    "id":315932,
    "subscriptionId":1646,
    "_domain":"carvoyant.com",
    "_type":"VEHICLECONNECTED",
    "_name":"VEHICLECONNECTED",
    "_timestamp":"20140912T010303+0000",
    "minimumTime":0,
    "httpStatusCode":200,
    "notificationPeriod":"INITIALSTATE",
    "dataSetId":4795435,
    "creatorClientId":"hasa2czfebhsj6XXXXXXXXXX",
    "vehicleId":123
  }
],
"totalRecords":2
}

```

Test EventSubscription

Generates an [EventNotification](#) for the specified [EventSubscription](#). This allows you to test that your subscription is configured properly. A dummy [EventNotification](#) object will be delivered to the `postUrl` provided in the requested [EventSubscription](#) and as a response to this request. The dummy notification will have null values where there would normally be recorded data that triggered the event.

Supported Verbs

- GET

GET

Query Paths

- `test/vehicle/{vehicle-id}/eventSubscription/{subscription-id}`

Query Parameters

Parameter	Description
vehicle-id	The Carvoyant identifier of the vehicle. This could be the device serial number in the car (for example, C201200001) or it could be the internal id returned from a previous lookup.
subscriptionid	The Carvoyant identifier of the subscription.

Call Options

Sortable	No
Pageable	No

Sample JSON Response:

```

{
  "notification": {
    "id": 1098702,

```

```
"subscriptionId": 1645,  
  "_domain": "carvoyant.com",  
  "_type": "LOWBATTERY",  
  "_name": "LOWBATTERY",  
  "_timestamp": "20141218T211057+0000",  
  "minimumTime": 0,  
  "httpStatusCode": 500,  
  "notificationPeriod": "STATECHANGE",  
  "dataSetId": null,  
  "creatorClientId": "hasa2czfebhsj6XXXXXXXXXX",  
  "thresholdVoltage": 12,  
  "recordedVoltage": null  
},  
"totalRecords": 1  
}
```

Sandbox API

4.1 Sandbox API

The Carvoyant Sandbox API provides a good way to build applications against the Carvoyant system without relying on a physical vehicle. This might be because you want to try out the Carvoyant platform before committing to a purchase, or simply because it's easier to develop against a known set of vehicle data. In either case, there are a few things you need to know to get started using it.

4.1.1 The Environment

The sandbox environment contains all elements of the production environment with the exception of the systems that collect data from actual vehicles on the road (and therefore doesn't support physical devices). Here are the endpoints that you need to get started:

<https://sandbox-driver.carvoyant.com>

This is the driver dashboard where you can register a Carvoyant account in the sandbox and create vehicles to test against.

<https://sandbox-api.carvoyant.com/sandbox/api>

This is the endpoint that you make API requests against. The resource space effectively matches production. Anywhere in our documentation where you see <https://api.carvoyant.com/v1/api> you would replace with <https://sandbox-api.carvoyant.com/sandbox/api>

<https://sandbox-auth.carvoyant.com>

This is our OAuth2 authorization endpoint where you can authorize your API to access your sandbox Carvoyant account.

<https://sandbox-simulator.carvoyant.com>

This is our traffic simulator for generating vehicle data in the sandbox environment.

4.1.2 Provisioning a Key

If you don't already have a developer account, you will need to register one here: <https://developer.carvoyant.com>

On the registration screen, make sure that the Sandbox API checkbox is selected:

Select which web APIs you will use

Issue a new key for Production API

Standard Developer

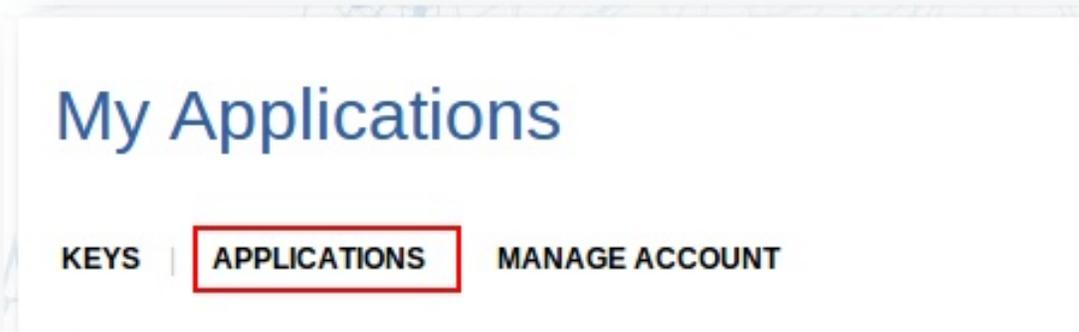
Key Rate Limits
10 Calls per second
5,000Calls per day

Issue a new key for Sandbox API

Sandbox Developer

Key Rate Limits
10 Calls per second
5,000Calls per day

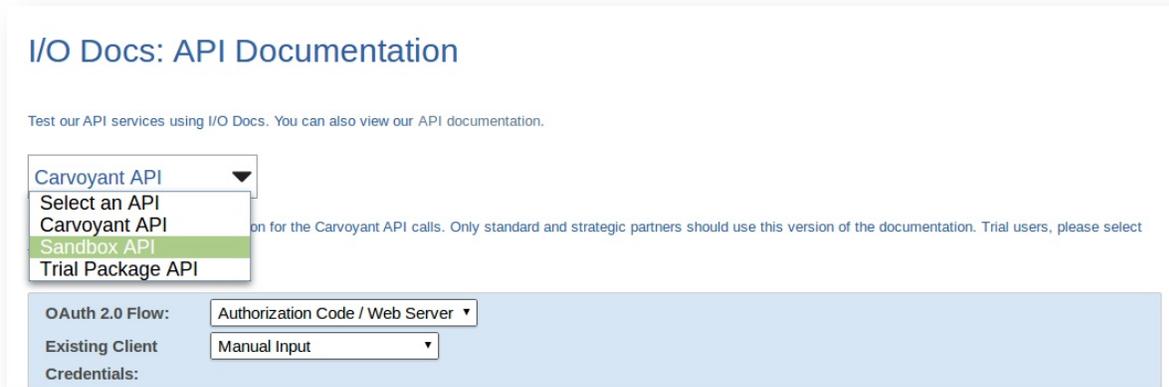
If you already have a developer account you can easily add a sandbox key to it as well. After logging in, select “My Account” from the top right area and then click the “Applications” heading item:



Next you need to decide whether you want to add the key to an existing application or create a new one. It’s probably easier to create a new one. Decide which you want to do and add the Sandbox API key.

4.1.3 Interactive API

Just like the production API, you can use the [online interactive API](#) to make calls against the Sandbox API. After creating your Sandbox API key, switch over to the Interactive API and change the API to Sandbox API.



4.1.4 Vehicle Data

Now that you're all set up to make calls against the Sandbox environment, you need some data! If you haven't already, add a vehicle to your Carvoyant account (or more than one - or more than one Carvoyant account, whatever you need). In the Sandbox environment, since there is no actual data collection, there are no hardware devices configured in the system so you will always leave the Device Id on the vehicle blank.

There are two ways that you can load up some data against your sandbox vehicles.

Inbound API

In the sandbox environment, we've opened up the ability for anyone to submit their own data through the API. This allows you to create very complex trips to meet whatever requirements you have for testing. An example of the data sets to submit in order to create a simple trip is below. You would pass each of these, in order, into the dataSet endpoint. All of them submitted would create a simple trip. The key is to ensure that the trip starts and contains only a single ignition ON event and ends with an ignition OFF event. Further, in order to track mileage correctly on the vehicle, you need to ensure that a GEN_TRIP_MILEAGE datum is in the ignition OFF message.

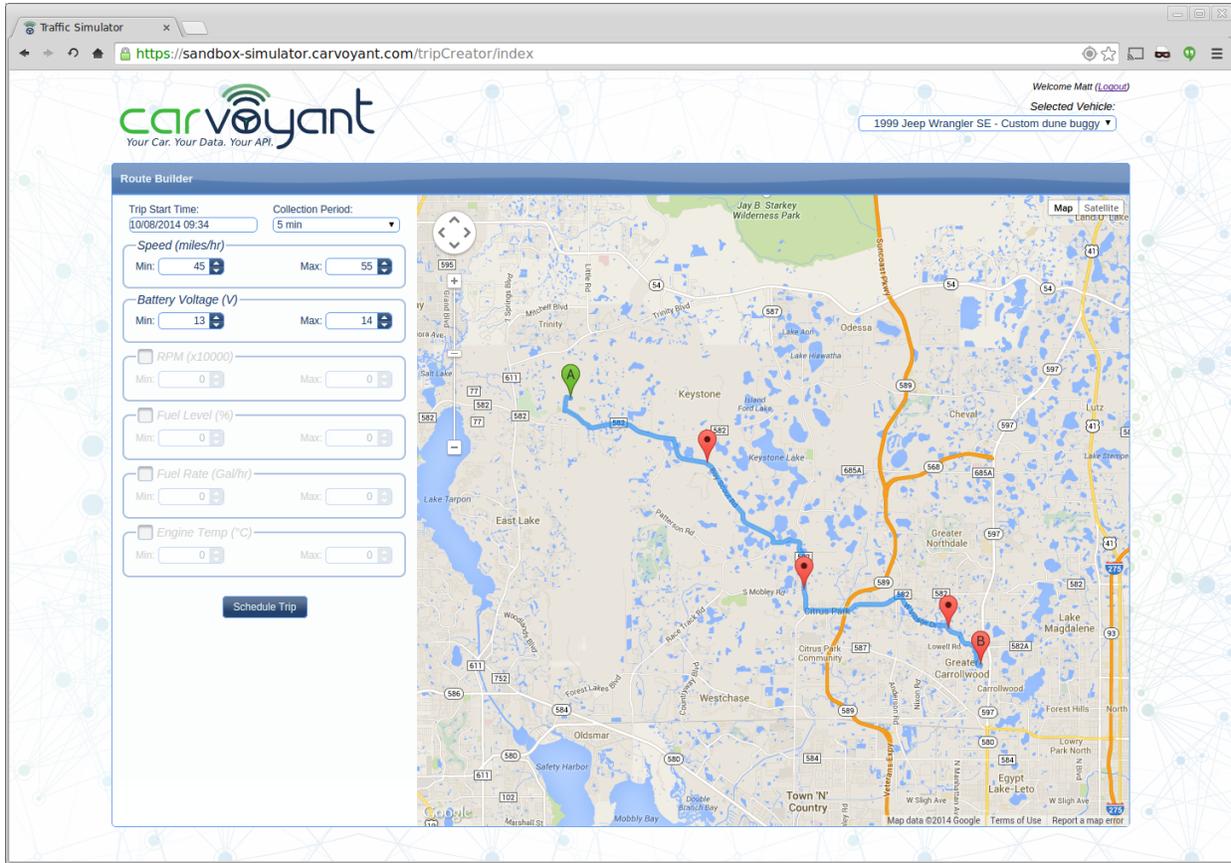
Sample JSON Messages:

```
{ "timestamp": "20140811T140444+0000", "ignitionStatus": "ON", "datum": [ { "timestamp": "20140811T140444+0000" } ] }
{ "timestamp": "20140811T140943+0000", "ignitionStatus": "RUNNING", "datum": [ { "timestamp": "20140811T140943+0000" } ] }
{ "timestamp": "20140811T141443+0000", "ignitionStatus": "RUNNING", "datum": [ { "timestamp": "20140811T141443+0000" } ] }
{ "timestamp": "20140811T141526+0000", "ignitionStatus": "OFF", "datum": [ { "timestamp": "20140811T141526+0000", "GEN_TRIP_MILEAGE": 1000 } ] }
```

Traffic Simulator

Having detailed access to submitting different data is nice, but sometimes you just want to simulate a couple of simple trips. To do that, we've created a simulation tool that allows you to select the vehicle you want, enter some points on a map, adjust some basic parameters if you want to, and then process the trip against your vehicle. This will send the data over just like a real vehicle would.

Open a web browser to <https://sandbox-simulator.carvoyant.com> and log in with your Carvoyant user credentials. Once you're logged in, set up a trip by clicking the start and end point on the map. When it's all set, click "Schedule Trip" and your trip will be sent to the Carvoyant platform.



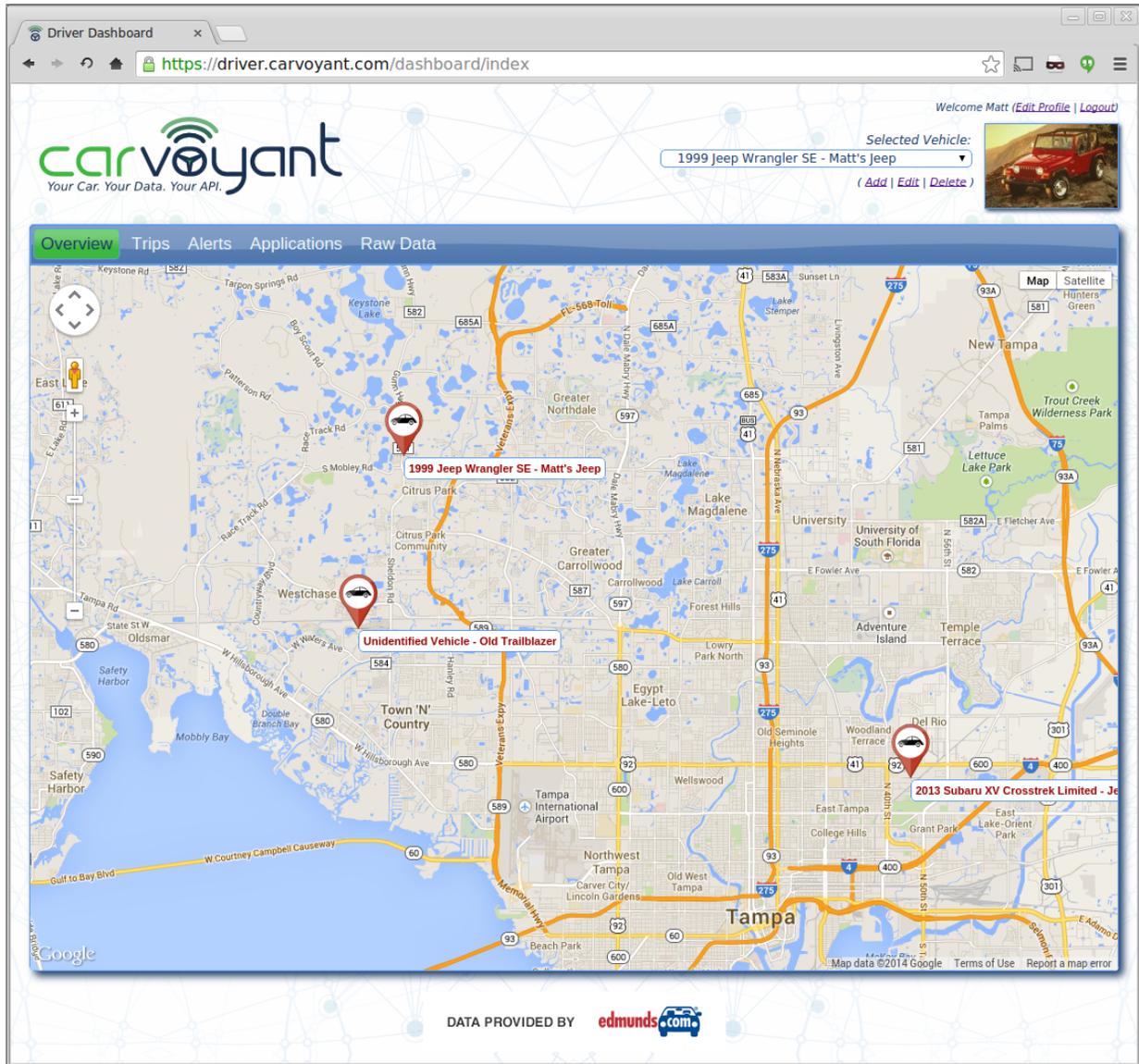
Driver Dashboard

5.1 Driver Dashboard

The Driver Dashboard application is a tool that Carvoyant users can use to register with our system and view basic information about their account (and it's associated vehicles). It can be accessed at <https://driver.carvoyant.com>.

5.1.1 Overview

When you first log in to the application, you will see a screen like this:



On the top right of this screen, you have a few different actions that can be taken. You can edit your users profile, change the currently selected vehicle, or manage the currently selected vehicle. Note that the vehicle that is selected will be the vehicle used to display data for on all of the other sections of the dashboard.

5.1.2 Trips

Within this section, you can review the information about all of the trips that you have taken. On the left will be every trip that Carvoyant has record of. When you select one, a map will display with the route of the trip. It will look like this. Note that if there were any events recorded for during the trip (such as a trouble code thrown), they will be displayed with an icon. Click the icon to see the details.

The screenshot shows the Carvoyant Driver Dashboard interface. At the top, there is a navigation bar with tabs for Overview, Trips, Alerts, Applications, and Raw Data. The 'Trips' tab is currently selected. On the left side, there is a 'Select Trip' table listing various trips with their start and end times. Below this table is a 'Trip Details' section for the selected trip. On the right side, there is a map view showing the location of the selected trip. The map includes a red location pin and a red line indicating the trip's path. At the bottom of the dashboard, there is a logo for 'DATA PROVIDED BY edmunds.com'.

Welcome Matt ([Edit Profile](#) | [Logout](#))

Selected Vehicle: 1999 Jeep Wrangler SE - Matt's Jeep ([Add](#) | [Edit](#) | [Delete](#))

Overview **Trips** Alerts Applications Raw Data

Select Trip

Start Time	End Time
Sep 4, 2014 7:13:34 AM	Sep 4, 2014 7:14:15 AM
Sep 4, 2014 7:02:51 AM	Sep 4, 2014 7:04:19 AM
Sep 3, 2014 7:33:50 PM	Sep 3, 2014 7:39:56 PM
Sep 3, 2014 6:20:25 PM	Sep 3, 2014 6:36:29 PM
Sep 3, 2014 1:02:35 PM	Sep 3, 2014 1:05:17 PM
Sep 3, 2014 12:29:55 PM	Sep 3, 2014 12:32:09 PM
Sep 3, 2014 9:34:01 AM	Sep 3, 2014 9:43:15 AM
Sep 3, 2014 7:38:22 AM	Sep 3, 2014 7:44:56 AM
Sep 3, 2014 7:13:53 AM	Sep 3, 2014 7:14:27 AM
Sep 3, 2014 7:02:53 AM	Sep 3, 2014 7:04:13 AM
Sep 2, 2014 5:03:08 PM	Sep 2, 2014 5:15:40 PM
Sep 2, 2014 9:56:39 AM	Sep 2, 2014 10:04:34 AM
Sep 2, 2014 7:34:47 AM	Sep 2, 2014 7:42:10 AM
Sep 2, 2014 7:30:00 AM	Sep 2, 2014 7:33:38 AM
Sep 2, 2014 7:13:56 AM	Sep 2, 2014 7:14:38 AM
Sep 2, 2014 7:03:07 AM	Sep 2, 2014 7:04:05 AM
Aug 29, 2014 7:11:27 AM	Aug 29, 2014 7:12:09 AM
Aug 29, 2014 7:04:50 AM	Aug 29, 2014 7:05:49 AM
Aug 28, 2014 7:13:09 AM	Aug 28, 2014 7:14:01 AM
Aug 28, 2014 7:03:25 AM	Aug 28, 2014 7:04:40 AM

Showing 21 to 40 of 2,121 entries

Previous 1 2 3 4 5 ... 107 Next

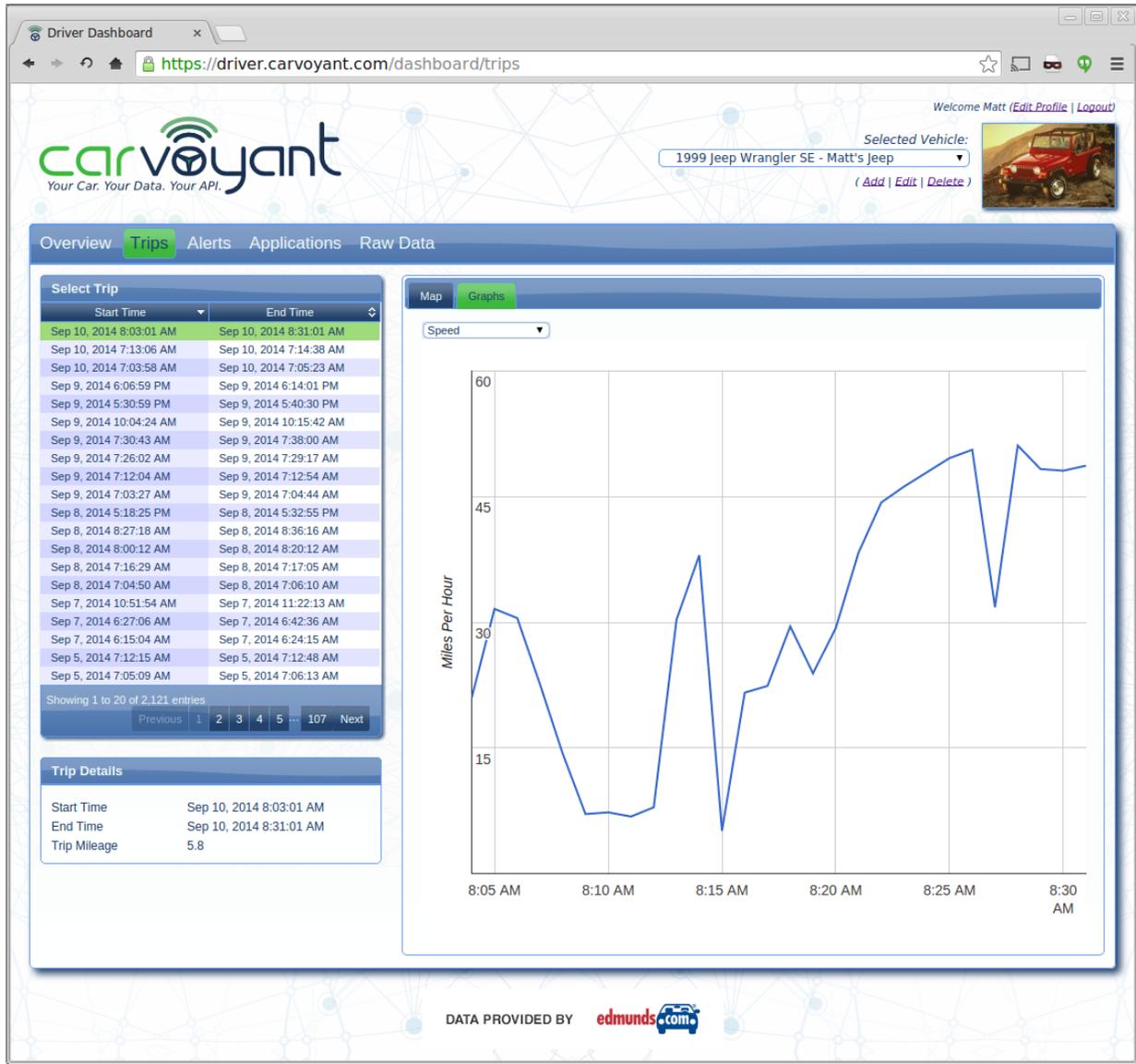
Trip Details

Start Time	Sep 2, 2014 7:30:00 AM
End Time	Sep 2, 2014 7:33:38 AM
Trip Mileage	1.2

Map Graphs

DATA PROVIDED BY edmunds.com

In addition to the map view, there is a Graphs tab that lets you see specific data for the trip:



5.1.3 Alerts

The Carvoyant system can notify users when different types of events occur. These are referred to as Alerts within the Driver Dashboard. This example has one ignition status alert set up for the vehicle:

Driver Dashboard

https://driver.carvoyant.com/dashboard/alerts

Welcome Matt ([Edit Profile](#) | [Logout](#))

Selected Vehicle: 1999 Jeep Wrangler SE - Matt's Jeep ([Add](#) | [Edit](#) | [Delete](#))

Overview Trips Alerts Applications Raw Data

Subscriptions Show Deleted?

Type	Created
IGNITIONSTATUS	Aug 26, 2014 7:19:44 PM

Showing 1 to 1 of 1 entries Previous Next

Subscription Details

Minimum Notification Interval	0 min
Notification Period	INITIALSTATE
Recipient Type: Glymipse	8133760384
Recipient Type: Email	matt@carvoyant.com

Create New Subscription Delete Selected Subscription

Notifications

Type	Created
IGNITIONSTATUS	Aug 26, 2014 7:37:39 PM
IGNITIONSTATUS	Aug 27, 2014 7:03:07 AM
IGNITIONSTATUS	Aug 27, 2014 7:12:12 AM
IGNITIONSTATUS	Aug 27, 2014 7:23:35 AM
IGNITIONSTATUS	Aug 27, 2014 9:36:06 AM
IGNITIONSTATUS	Aug 27, 2014 9:47:37 AM
IGNITIONSTATUS	Aug 27, 2014 3:38:41 PM
IGNITIONSTATUS	Aug 28, 2014 7:03:29 AM
IGNITIONSTATUS	Aug 28, 2014 7:13:16 AM
IGNITIONSTATUS	Aug 29, 2014 7:04:56 AM
IGNITIONSTATUS	Aug 29, 2014 7:11:40 AM
IGNITIONSTATUS	Sep 2, 2014 7:03:11 AM
IGNITIONSTATUS	Sep 2, 2014 7:14:05 AM
IGNITIONSTATUS	Sep 2, 2014 7:30:04 AM
IGNITIONSTATUS	Sep 2, 2014 7:34:50 AM

Showing 1 to 15 of 47 entries Previous Next

DATA PROVIDED BY edmunds.com

After selecting a subscription, you will see a list of any notifications that have been sent for it. Selecting a notification will display details of the alert.

The following alerts can be created:

- GeoFence - This alert will let you monitor the location where your vehicles are travelling.
- Ignition Status - This alert monitors whether the vehicle is on or off.
- Low Battery - This alert indicates that the battery voltage has dropped below 12.0 volts.
- Numeric Data Key - This alert lets you monitor any numeric data element (for example, when engine temperature has gone over a certain value).
- Time of Day - This alert lets you monitor the time of day that a vehicle is being used. Note that the alert checks for vehicle activity outside the specified values.
- Trouble Code - This alert monitors for any trouble codes that the vehicle throws.

When configuring an alert, the following properties can be selected for each alert type:

	Ge-oFence	Ignition Status	Low Battery	Numeric Data Key	Time of Day	Trouble Code
Minimum Notification Interval	X	X	X	X	X	X
Notification Period	X	X	X	X	X	X
Recipients	X	X	X	X	X	X
Location Map	X					
Radius	X					
Ignition Status	X					
Boundary Condition	X					
Data Key				X		
Threshold Value				X		
Relationship				X		
Start Time					X	
End Time					X	
Days of Week					X	

Property Descriptions

Minimum Notification Interval

This is the minimum time between alerts in minutes. If you set this value to 60, you will only receive one alert every 60 minutes even if the event occurs more frequently.

Notification Period

This describes when the alert will be checked. Possible values are:

- Initial State - Only when the conditions are first met
- One Time - Only when the conditions are first met, and then the alert is deleted (subsequent events will not alert you)
- Continuous - Alerts will be sent every time the Carvoyant system collects data that meet the alerts criteria
- State Change - Alerts will be sent when the criteria is first met, and then when the criteria is no longer met

Recipients

The recipients of the alerts. Currently we support Email and Glympse. Email will send a simple email to the addresses listed. Glympse is an integration with the Glympse service and is currently in beta.

Location Map

Select a location on the map.

Radius

The distance from a point on the map. This is in miles but can be partial (for example, 0.2).

Ignition Status

This can be one of three values:

- On - When the vehicle is first turned on
- Off - When the vehicle is turned off
- Running - Any time the vehicle is running

Boundary Condition

This can be one of two values:

- Inside: The geofence is triggered when the vehicle is within the radius of the specified location
- Outside: The geofence is triggered when the vehicle is outside the radius of the specified location

Data Key

Specified what data key type to monitor.

Threshold Value

The value of the recorded data key to check against.

Relationship

This can be one of three values:

- Above - The recorded data is above the threshold value
- Below - The recorded data is below the threshold value
- Equals - The recorded data equals the threshold value

Start Time

The start time of the allowed window.

End Time

The end time of the allowed window.

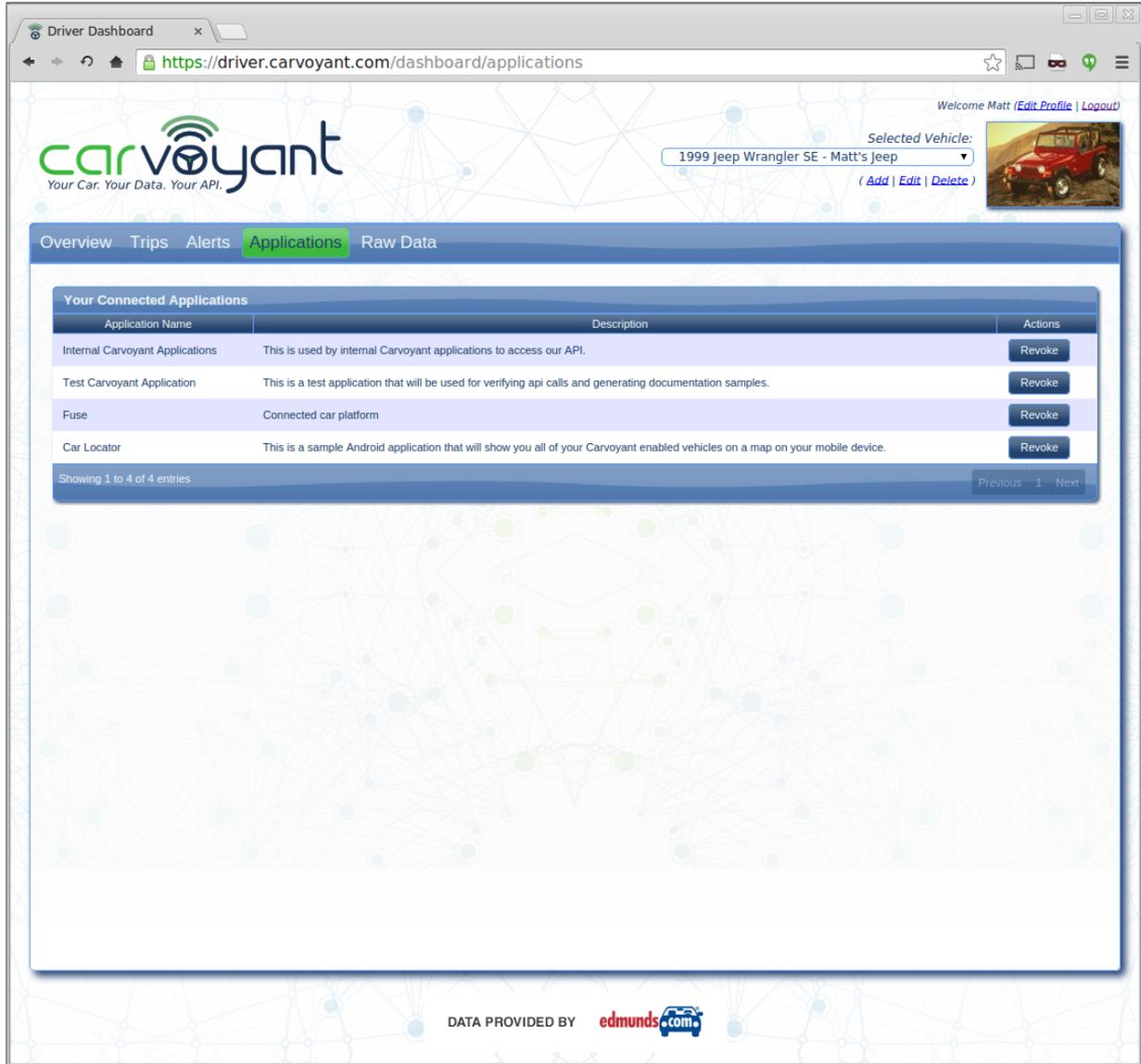
Days of Week

The days of the week for the allowed window.

5.1.4 Applications

Most Carvoyant users will end up having multiple applications that are used to interact with your car. The Carvoyant system provides that connectivity to your car and a means for other application developers to interact with it. This page will allow you to review who you have granted access to your car, and if necessary, revoke that access.

Please note that if you revoke access to an application, that application will no longer work until you reauthorize it from within that applications.



5.1.5 Raw Data

The Raw Data page will allow you to see every piece of data that Carvoyant has collected for the current vehicle. It is sortable by date and can be filtered by the type of data that you are interested in.

The screenshot shows the Carvoyant Driver Dashboard interface. At the top, the Carvoyant logo is displayed with the tagline "Your Car. Your Data. Your API." The user is logged in as "Matt" and is viewing the "Raw Data" section for a "1999 Jeep Wrangler SE - Matt's Jeep". The dashboard features a navigation menu with "Overview", "Trips", "Alerts", "Applications", and "Raw Data".

The main content area displays a table titled "Raw Vehicle Data" with a dropdown menu set to "GEN_ENGINE_COOLANT_TEMP". The table contains the following data:

Timestamp	Data Type	Value
Sep 10, 2014 8:31:01 AM	GEN_ENGINE_COOLANT_TEMP	197.6 deg F
Sep 10, 2014 8:31:00 AM	GEN_ENGINE_COOLANT_TEMP	197.6 deg F
Sep 10, 2014 8:30:00 AM	GEN_ENGINE_COOLANT_TEMP	206.6 deg F
Sep 10, 2014 8:29:00 AM	GEN_ENGINE_COOLANT_TEMP	199.4 deg F
Sep 10, 2014 8:28:00 AM	GEN_ENGINE_COOLANT_TEMP	201.2 deg F
Sep 10, 2014 8:27:00 AM	GEN_ENGINE_COOLANT_TEMP	206.6 deg F
Sep 10, 2014 8:26:00 AM	GEN_ENGINE_COOLANT_TEMP	192.2 deg F
Sep 10, 2014 8:25:00 AM	GEN_ENGINE_COOLANT_TEMP	206.6 deg F
Sep 10, 2014 8:24:00 AM	GEN_ENGINE_COOLANT_TEMP	208.4 deg F
Sep 10, 2014 8:23:00 AM	GEN_ENGINE_COOLANT_TEMP	199.4 deg F
Sep 10, 2014 8:22:00 AM	GEN_ENGINE_COOLANT_TEMP	197.6 deg F
Sep 10, 2014 8:21:00 AM	GEN_ENGINE_COOLANT_TEMP	197.6 deg F
Sep 10, 2014 8:20:00 AM	GEN_ENGINE_COOLANT_TEMP	204.8 deg F
Sep 10, 2014 8:19:00 AM	GEN_ENGINE_COOLANT_TEMP	199.4 deg F
Sep 10, 2014 8:18:00 AM	GEN_ENGINE_COOLANT_TEMP	204.8 deg F
Sep 10, 2014 8:17:00 AM	GEN_ENGINE_COOLANT_TEMP	199.4 deg F
Sep 10, 2014 8:16:00 AM	GEN_ENGINE_COOLANT_TEMP	199.4 deg F
Sep 10, 2014 8:15:00 AM	GEN_ENGINE_COOLANT_TEMP	199.4 deg F
Sep 10, 2014 8:14:00 AM	GEN_ENGINE_COOLANT_TEMP	197.6 deg F
Sep 10, 2014 8:13:00 AM	GEN_ENGINE_COOLANT_TEMP	201.2 deg F
Sep 10, 2014 8:12:00 AM	GEN_ENGINE_COOLANT_TEMP	208.4 deg F
Sep 10, 2014 8:11:00 AM	GEN_ENGINE_COOLANT_TEMP	204.8 deg F
Sep 10, 2014 8:10:00 AM	GEN_ENGINE_COOLANT_TEMP	195.8 deg F
Sep 10, 2014 8:09:00 AM	GEN_ENGINE_COOLANT_TEMP	192.2 deg F
Sep 10, 2014 8:08:00 AM	GEN_ENGINE_COOLANT_TEMP	192.2 deg F
Sep 10, 2014 8:07:00 AM	GEN_ENGINE_COOLANT_TEMP	192.2 deg F
Sep 10, 2014 8:06:00 AM	GEN_ENGINE_COOLANT_TEMP	194.0 deg F
Sep 10, 2014 8:05:00 AM	GEN_ENGINE_COOLANT_TEMP	179.6 deg F
Sep 10, 2014 8:04:00 AM	GEN_ENGINE_COOLANT_TEMP	150.8 deg F
Sep 10, 2014 7:14:38 AM	GEN_ENGINE_COOLANT_TEMP	159.8 deg F

At the bottom of the table, it indicates "Showing 1 to 30 of 18,818 entries (filtered from 30 total entries)". A pagination control shows "Previous", "1", "2", "3", "4", "5", "628", and "Next".

At the bottom of the dashboard, it says "DATA PROVIDED BY edmunds.com".

Sample Integrations

6.1 Sample Integrations

In this section we will periodically post how-to's for connecting the Carvoyant platform to other systems.

All of these integrations will use the Carvoyant Sandbox environment. They all expect that the developer has created a Developer account, a Sandbox API key and a Carvoyant Sandbox User Account (with a vehicle on it). Instructions for getting started can be found [here](#). Please be sure that your Sandbox environment is configured properly and that you are able to make API calls against the Carvoyant account.

6.1.1 scriptr.io

[scriptr.io](#) is a secure Cloud based platform for implementing the back-end of IoT applications. It provides developers and development teams with a web IDE that integrates with code repositories, and a wide range of native APIs, SDKs and connectors that dramatically simplify development and reduce time to market. [scriptr.io](#) also takes care of your application's infrastructure needs such as security, scalability and reliability. In addition, developers can get access to enterprise connectors through [scriptr.io](#)'s partners such as [wot.io](#). The [scriptr.io](#) team has prepared a tutorial for building Carvoyant applications on the [scriptr.io](#) platform. You can view the tutorial [here](#).

6.1.2 SmartThings

This example integration will add your Carvoyant enabled vehicles to your SmartThings setup. You will be able to integrate your vehicles presence as well as it's ignition state into your smart home configuration. View the how-to [here](#).

6.1.3 Force.com

This example integration will use Vehicle location Event Subscriptions to update a map within a Force.com application. View the how-to [here](#).