Oregon Department of Transportation

Intelligent Transportation Systems

TripCheck API

Getting Started Guide

Description

The TripCheck API is designed to provide developers with access to the data available on ODOT's traveler information website, including incidents, cameras, message signs, weather stations, and more. With the TripCheck API, you can use ODOT's data endpoints to develop your own integrated applications, like TripCheck, TripCheck for Twitter, or TripCheck TV. The purpose of this document is to assist end users in the utilization of the TripCheck API and web portal.

History

Version	Description of Change	Author	Date
1.0	Document creation	ODOT	
2.0	Document restructured	ODOT	11/13/19
3.0	Document updated	ODOT	11/19/20
4.0	Document updated	ODOT	05/12/20
5.0	Document updated with new screenshots and datafeed documentation	ODOT	06/30/20

Contents

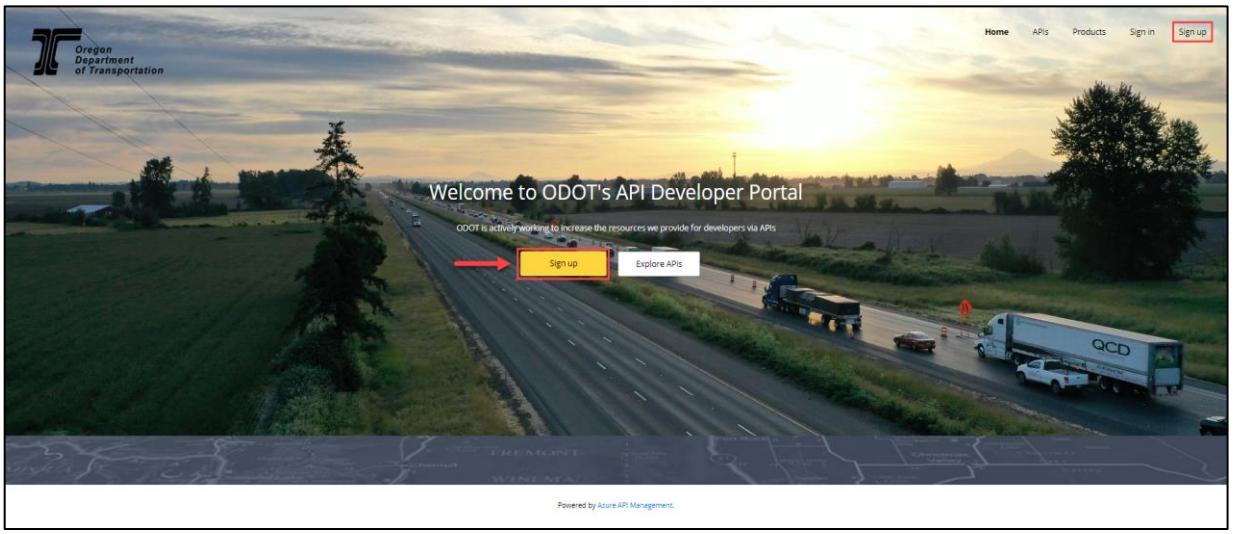
1. ODOT Developer Portal and API Getting Started Guide

1.1 Developer Portal Sign up Process

An end user is not required to sign up for ODOT's developer portal to be able view the portal. Signing up for the portal allows an end user to be able to obtain an API subscription key, and subsequently view ODOT's traveler information data in XML and JSON markup languages.

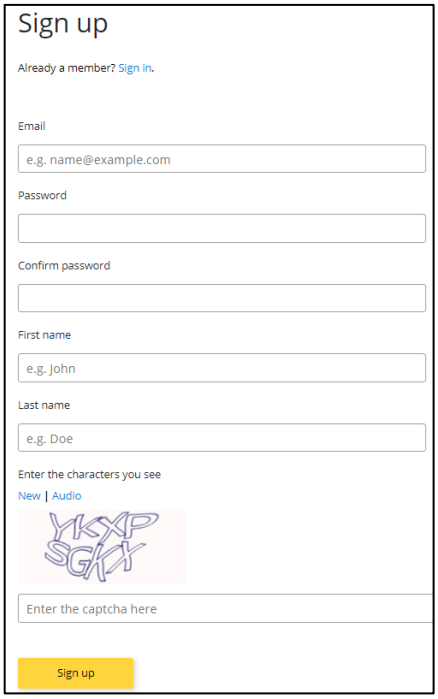
1.1.1 Step 1

There are different methods of navigating to the 'Sign up' page, the simplest being **selecting the 'Sign up' button** on the 'Home' tab:



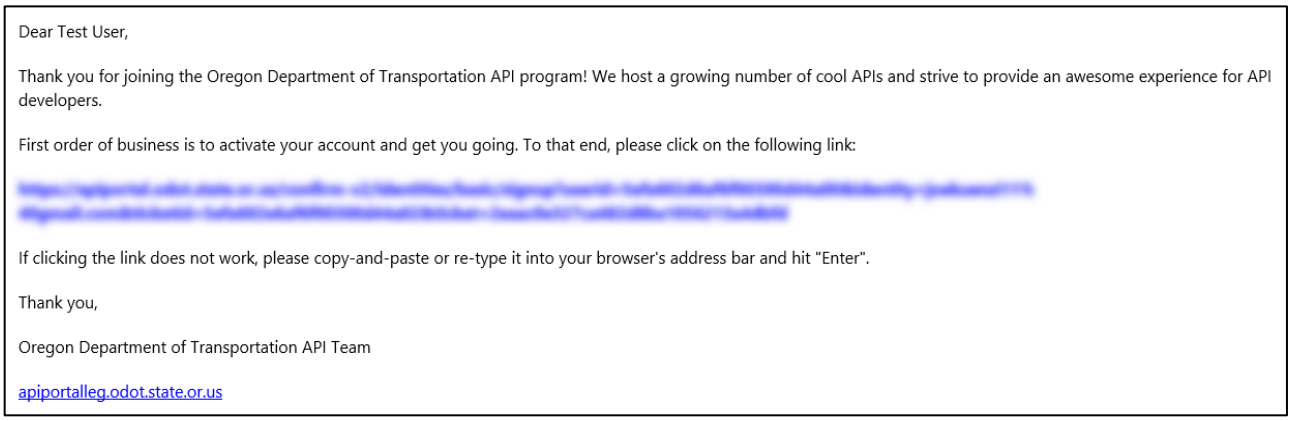
1.1.2 Step 2

Once the 'Sign up' page opens, **input the required information** into the form textboxes and **select 'Sign up'**



1.1.3 Step 3

Once ‘Sign up’ is selected, an email is sent to the email addressed provided on the ‘Sign up’ page. To validate your account, **select the link provided in the email**



Dear Test User,

Thank you for joining the Oregon Department of Transportation API program! We host a growing number of cool APIs and strive to provide an awesome experience for API developers.

First order of business is to activate your account and get you going. To that end, please click on the following link:

[redacted]

If clicking the link does not work, please copy-and-paste or re-type it into your browser's address bar and hit "Enter".

Thank you,

Oregon Department of Transportation API Team

apiportalleg.odot.state.or.us

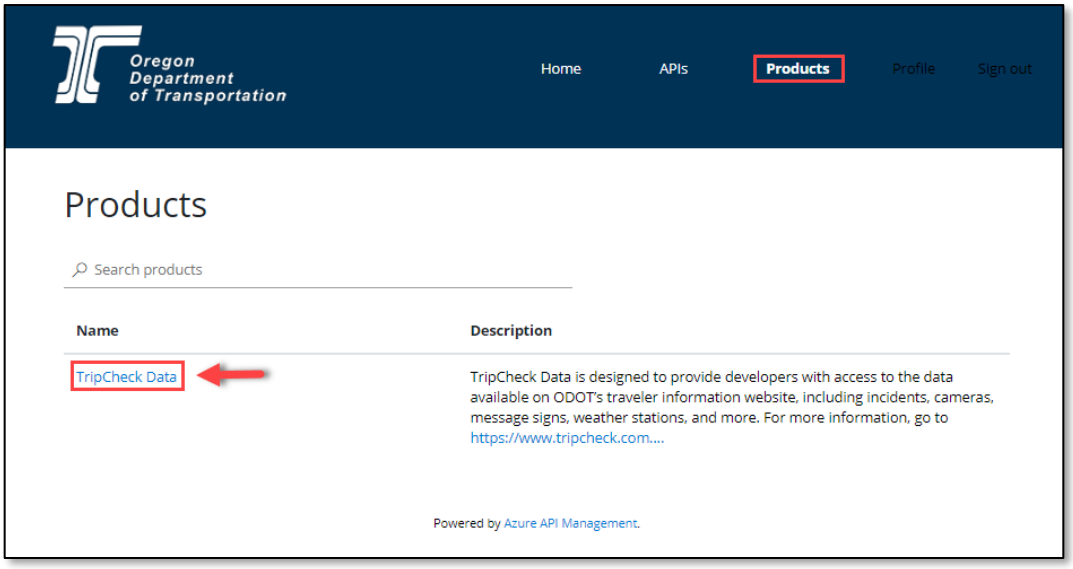
Once the account activation link has been selected, the user’s account profile is opened within the web portal and a welcome email containing links to resources is sent to the users email address.

1.2 TripCheck API Subscription

Once a valid web portal account is created, a user can obtain a unique API access key that will be used to retrieve desired datasets from the TripCheck API. For API subscription help follow the steps listed below:

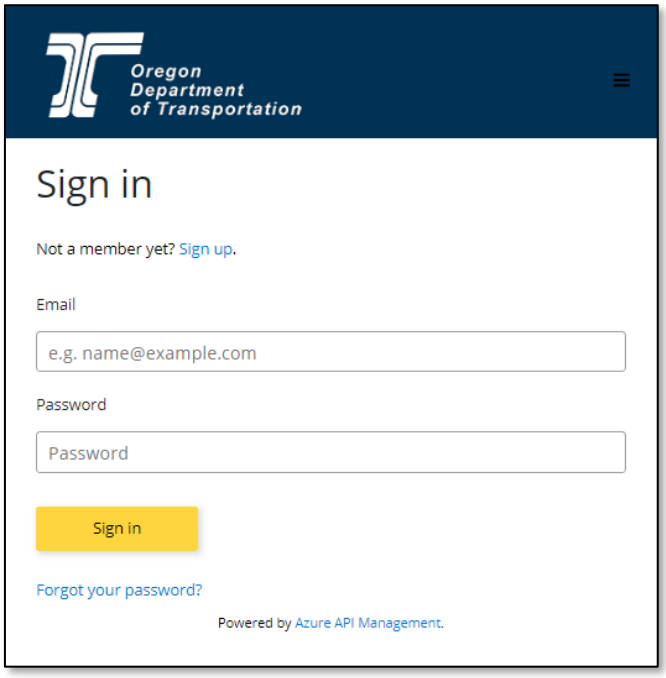
1.2.1 Step 1

To get started using the TripCheck API from the Developer Portal, **navigate to the 'Products' tab** then **select the 'TripCheck Data' link.**



1.2.2 Step 2

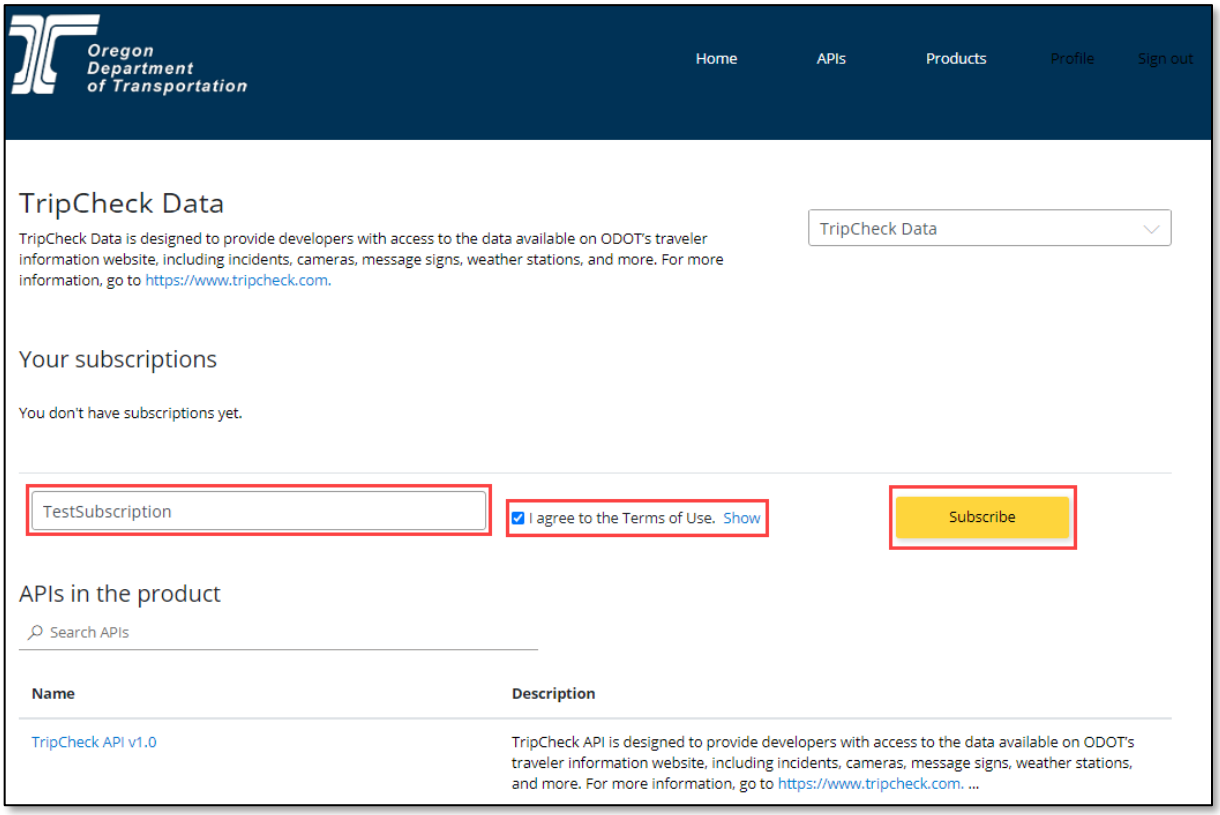
If the user **is not** signed into the web portal, the 'Sign in' page will display for the user to enter their login credentials:



1.2.3 Step 3

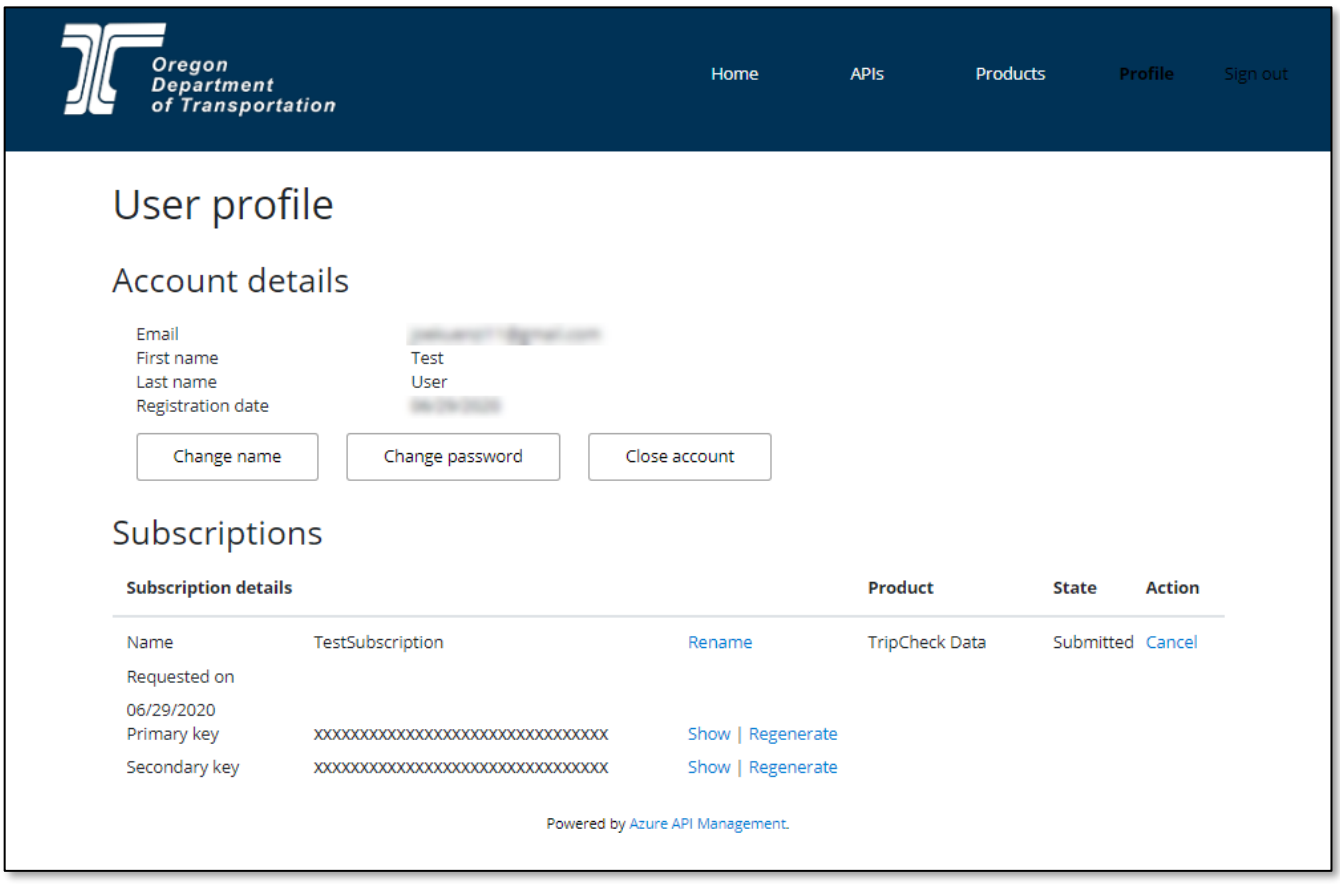
While signed in, The 'TripCheck Data' page contains helpful information regarding the traveler data available via the API. Subscribing will provide a valid user with an API subscription key. To obtain a subscription key, follow the steps below:

1. Input a **product subscription name** of your choice
2. **Read the Terms of Use by selecting the 'Show' button** and if agreed upon, confirm that they are acceptable by selecting the **"I agree" radio button.**
3. Finally, **select the 'Subscribe' button** in the right-hand corner of the page to subscribe to the TripCheck API:

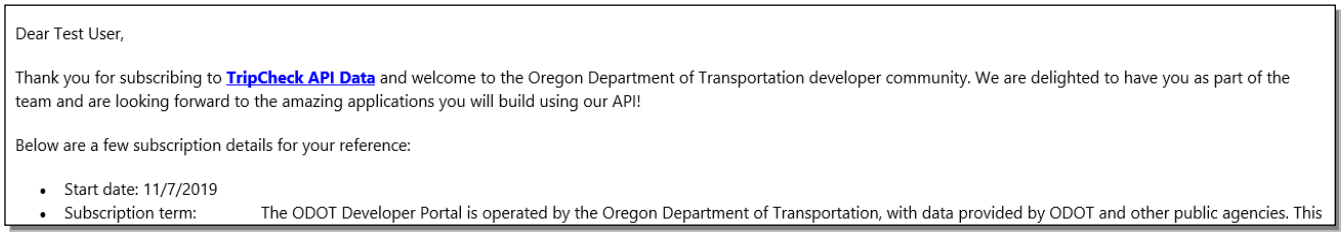


1.2.4 Step 4

Once the user account is subscribed to the API, a unique access key is assigned to the user. This access key is visible on the 'User Profile' *(Click the 'Show' link to expose your key).*



Also a confirmation email is sent to the email address provided that contains details and helpful hints for utilizing the API:



Dear Test User,

Thank you for subscribing to **TripCheck API Data** and welcome to the Oregon Department of Transportation developer community. We are delighted to have you as part of the team and are looking forward to the amazing applications you will build using our API!

Below are a few subscription details for your reference:

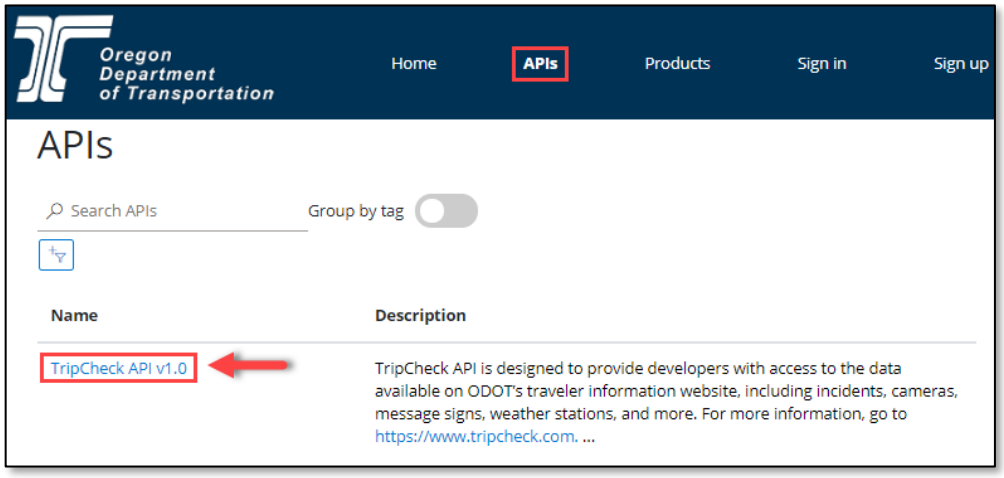
- Start date: 11/7/2019
- Subscription term: The ODOT Developer Portal is operated by the Oregon Department of Transportation, with data provided by ODOT and other public agencies. This

1.3 Using the Portal to view Datasets

Once an account has successfully signed up to use the portal, and subscribed to the TripCheck API, then the ability to view ODOT's traveler information data in XML/JSON is available. The steps below detail how to view ODOT's traveler information data from the web portal.

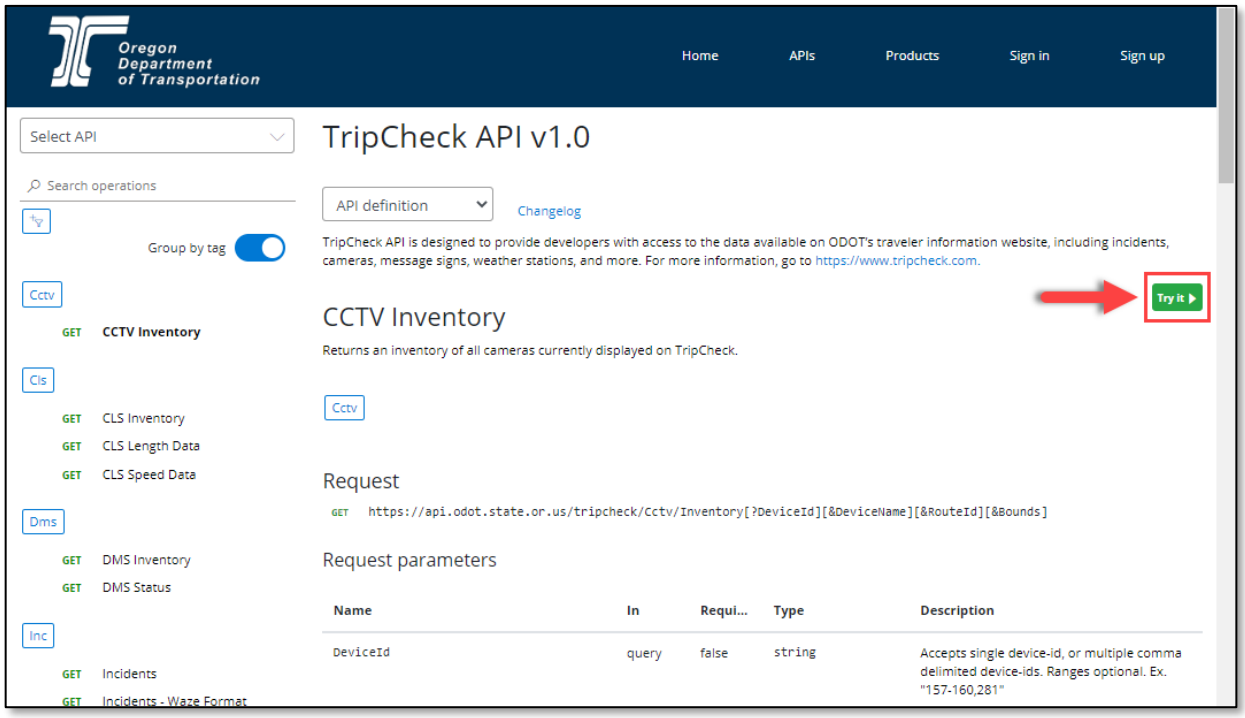
1.3.1 Step 1

To view data offered up by the TripCheck API from the Developer Portal, **navigate to the 'APIs' tab** then **select the 'TripCheck API v1.0' link**



1.3.2 Step 2

Select a datafeed of interest in the column on the left-hand side of the page. Datafeed descriptions, definitions, and sample responses are listed to provide additional context and background on the contents and structure of the datafeed of interest.
Select the 'Try it' button *(if user is not signed in, or not subscribed, then a '401 Access Denied' response will be returned)*

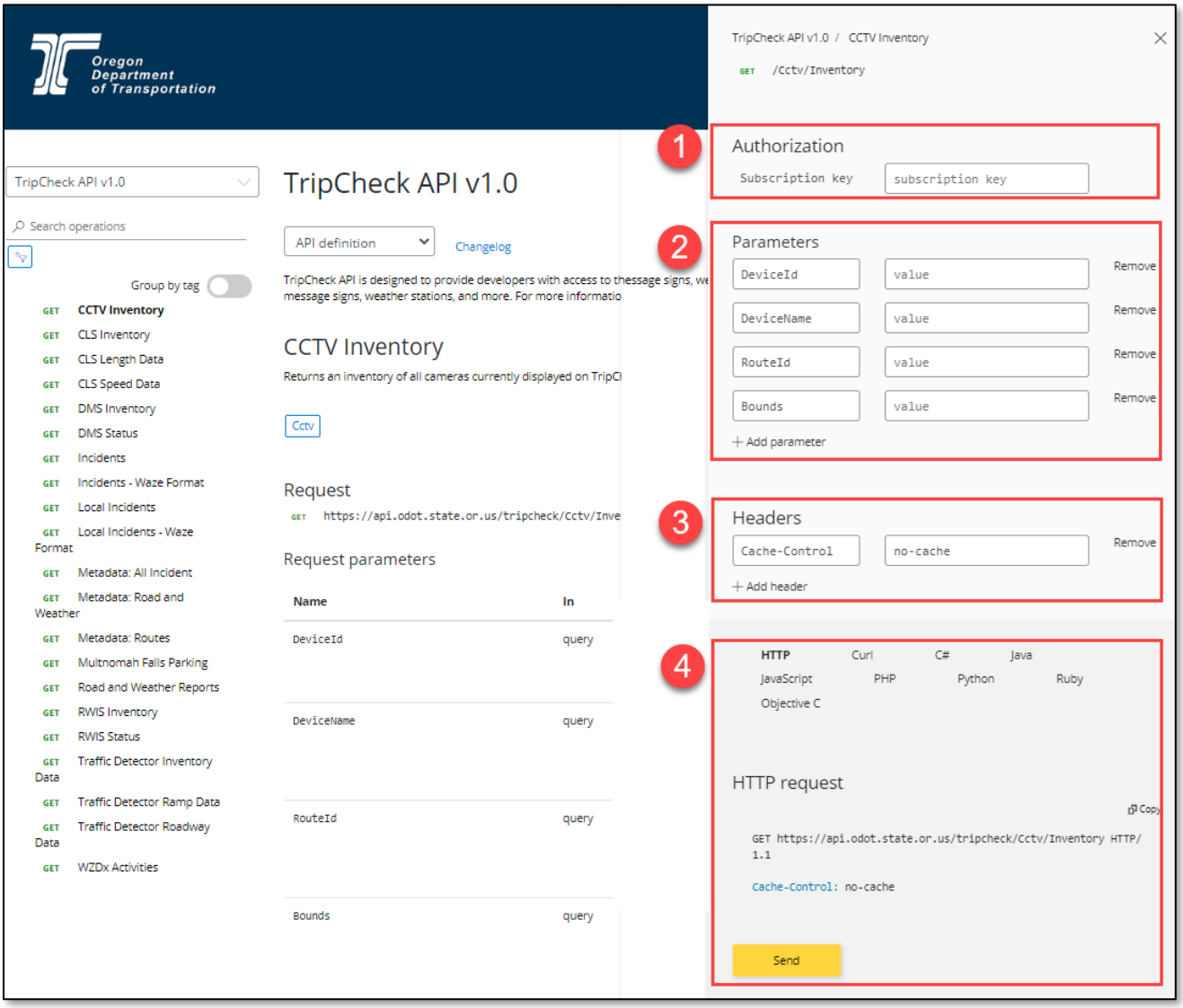


1.3.3 Step 3

After selecting the 'Try it' button, the sections in the graphic below are displayed:

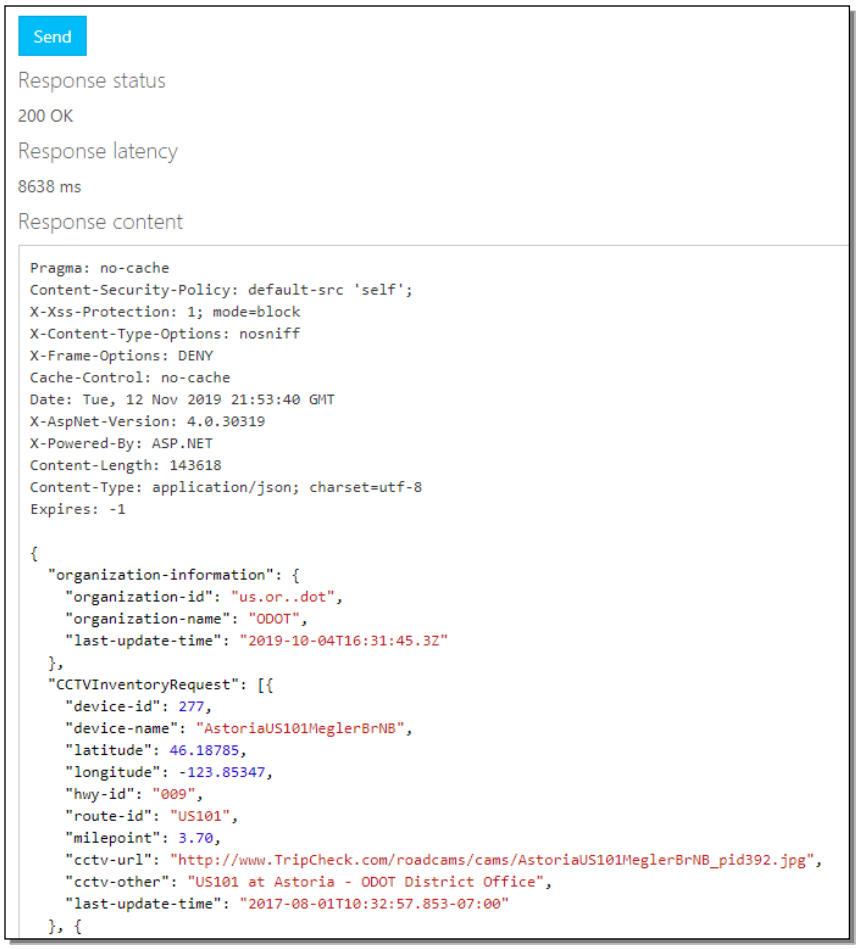
1. **Authorization:** Lists both the primary and secondary subscription keys
2. **Dataset parameters:** Allows entry of appropriate parameter value(s) that pair down the requested dataset into the HTTP request and Request URL. For more detailed and specific details regarding query parameters, see section ***2.4 Dataset Request Parameters***.
3. **Headers:** Allows entry of appropriate values into HTTP request and Request URL. The subscription key is auto-populated as a header to allow the request access to the dataset. This is also the section where the data format can be specified in the request by including the 'Accept' header with the data format needed. The available data formats for request are: application/json, text/json, application/xml, and text/xml.
4. **Request URL and HTTP request:** The components of the request that is sent to the API to retrieve Traveler information data.

Select the 'Send' button at the bottom of the page to send the request to retrieve the dataset of interest.



1.3.4 Step 4

After sending an appropriately formatted data request, the dataset of interest is returned:

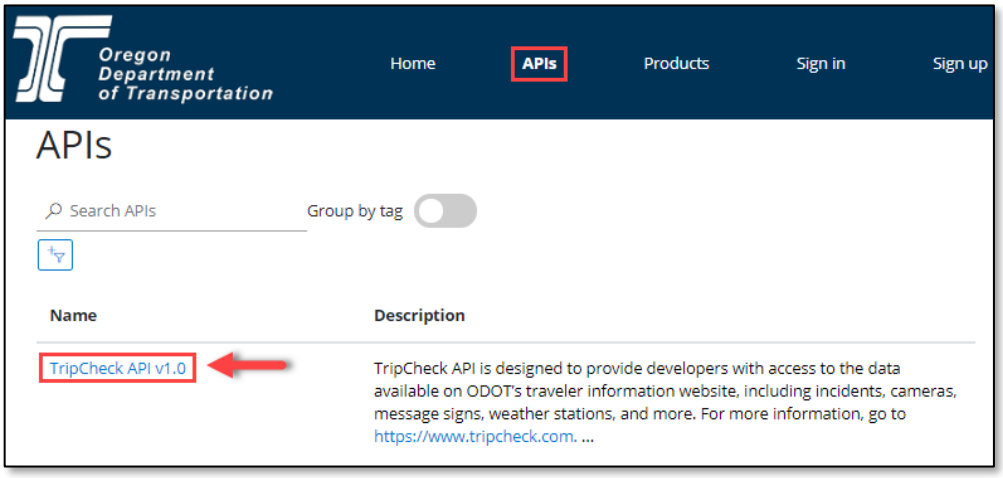


1.4 Retrieving Datasets via the API

Once an account has successfully signed up to use the portal, and subscribed to the TripCheck API, then the ability to request and retrieve ODOT's traveler information data via an API call is available. The steps below detail some possible methods to request and retrieve ODOT's traveler information data via an API call.

1.4.1 Step 1

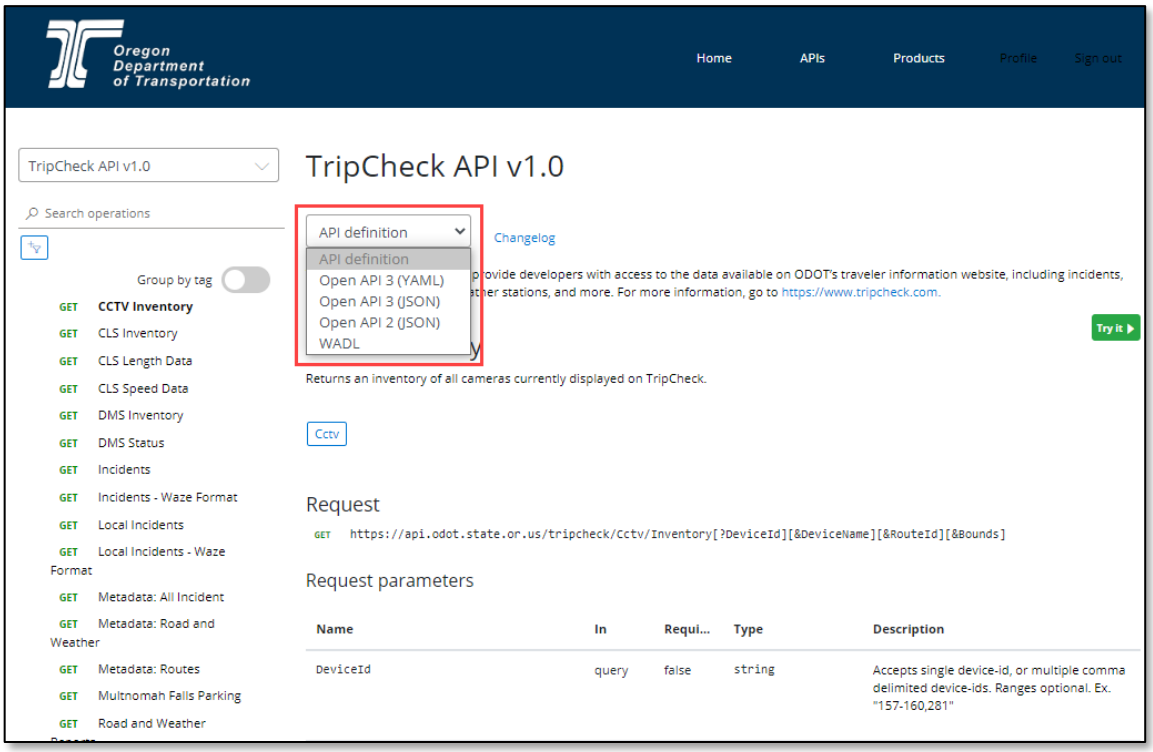
To get additional details on how to request and retrieve datasets available via the TripCheck API, **navigate to the 'APIs' tab** then **select the 'TripCheck API' link**



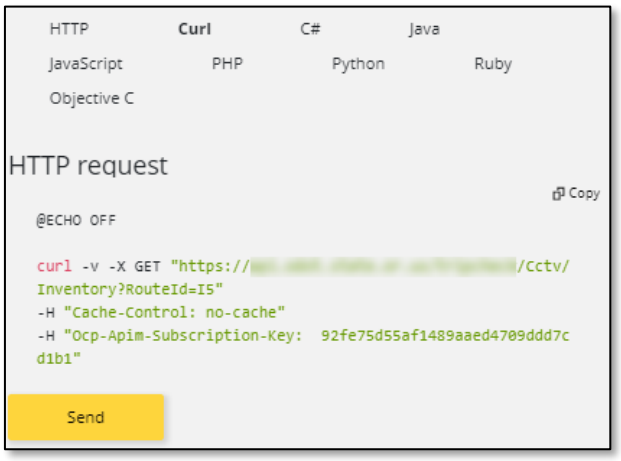
1.4.2 Step 2

Select a datafeed of interest in the column on the left-hand side of the page. Datafeed descriptions, definitions, and samples are displayed to provide more context and background on the datafeed of interest.

Select 'API definition' to find resources such as the WADL model that gives further details regarding the web service.



Navigate to the bottom of the page to view the request samples section of the page. This section gives code samples of multiple different coding language requests to retrieve datasets from the TripCheck API.



You're all set! With the TripCheck Data API, you can use ODOT's data endpoints to develop your own integrated applications, websites, and mobile apps. Happy programming!

2. ODOT Developer Portal and API Additional Information

2.1 Available Data Feeds and Refresh Rates

Data Feed	Resource	Refresh Rate	Description
Cameras	CCTV Inventory	24 hours	The CCTV Inventory datafeed provides an inventory of all available cameras currently displayed on TripCheck, along with an Internet URL that can be used to access the specific still camera image. Cameras may be ODOT owned and maintained or owned and maintained by a partner agency.
Classified Length and Speed	CLS- Inventory	10 minutes	Vehicle Length and Speed classifications for each Detector Station. This data defines what vehicle length values are being used for aggregation. The data is collected through web services on Automated Traffic Controllers (ATCs).
	CLS - Length	10 minutes	Length data aggregated by length classification for each detector station. The Bin Count represents the number of vehicles that passed the detector station in a 20 second period that fall into that particular length classification.
	CLS - Speed	10 minutes	Speed data aggregated by speed classification for each detector station. The Bin Count represents the speed of vehicles that passed the detector station in a 20 second period that fall into that particular speed classification.
Dynamic Message Signs (DMS)	DMS Inventory	24 hours	Name and Location of each Dynamic Message Sign on state highways or controlled by ODOT.
	DMS Status	1 minute	Statewide Dynamic Message Sign (DMS) returns current message data for active signs in the State of Oregon.
Incidents	Incidents	30 seconds	Current traffic incidents that are being reported on State Highways by ODOT – e.g. crashes, planned closures, and construction zones.
	Incidents: Waze Format	30 seconds	Current traffic incidents that are being reported on state highways by ODOT and formatted to the Waze CIFS V2 standard – e.g. crashes, planned closures, and construction zones.
	Metadata: All Incident	24 hours	Returns an inventory of the enumerated values that are held within the TripCheck API Incidents and TLE Incidents datafeeds.
Local Incidents	Local Incidents	2 minutes	Events occurring on local and county roads as reported by non-ODOT government agencies (ex., Washington County, City of Eugene Public Works).
	Local Incidents: Waze Format	2 minutes	Events occurring on local and county roads as reported by non-ODOT government agencies and formatted to the Waze CIFS V2 standard… (ex., Washington County, City of Eugene Public Works).
Metadata: Routes	Metadata: Routes	24 hours	Returns a list of route names currently in the system
Multnomah Falls Parking	Multnomah Falls Parking	20 seconds	Parking lot occupancy and Gate closure data for the Multnomah Falls parking lot at Exit 31 of I-84.

Road & Weather	Road and Weather Reports	5 minutes	Current road conditions as reported by ODOT crews. This includes weather observations and tire chain restrictions.
	Metadata: Road and Weather	24 hours	Returns an inventory of the enumerated values that are held within the TripCheck API Road and Weather datafeed.
Roadside Weather Information Systems (RWIS)	RWIS Inventory	24 hours	Name and location of all the Weather Stations along state highways and what each station can measure.
	RWIS Status	5 minutes	Weather data from automated Weather stations along state highways (e.g. Air Temperature, Surface Temperature, wind speed, etc.) Note: not all stations can measure all types of weather factors.
Traffic Detectors	Traffic Detector: Inventory	24 hours	Name and location of traffic detector stations and highway ramps associated with them.
	Traffic Detector: Roadway Data	2 minutes	Roadway traffic detectors collecting volume, occupancy and speed data from select roadways located in Oregon.
	Traffic Detector: Ramp Data	2 minutes	Highway ramp data such as ramp occupancy, volume, and metering rate collected by ODOT Central Ramp Metering System for select ramps located in Oregon.
Work Zone Data Exchange	WZDx Activities	30 seconds	Work zone related activities occurring throughout the State of Oregon formatted according to the WZDx standard created by the FHWA and USDOT.

2.2 API Response Codes

Response Code	Status Message	Reason for Status
200	**OK**	Datafeed is available and the dataset was returned successfully
401	**Access Denied**	Invalid or missing API subscription key
400	**Bad Request**	Invalid parameter
404	**Not Found**	CCTV inventory not available
429	**Too Many Requests**	Message request rate limit exceeded

2.3 Dataset Request Parameters

2.3.1 Dataset Request Parameters High Level Overview

The TripCheck API allows a subscriber to input parameters to filter and retrieve a certain subset of data. For example, a subscriber can filter by a route to receive only the data subset occurring on specific routes. There is also the possibility for a subscriber to enter an invalid parameter value. The purpose of this section is to provide a high-level overview of the process the TripCheck API will use to validate whether the parameter value inputted by the subscriber is valid or invalid, and what error handling will occur when invalid.

The Data Portal parameter validation process performs the following functions:

a) Examines the inputted parameter value(s)
b) Validates that all inputted parameter value(s) contains at least one corresponding value stored within a metadata feed, or inventory feed inside the memory cache. For parameters that are not contained within either a metadata or inventory datafeed, a search is performed to find if the value inputted has a match within the datafeed in question; (e.g. unique identifiers etc…)

- For example, if an external process is requesting the DMS_Inventory datafeed parameterized with route-id of 'US', the service will validate that within the *Metadata: Routes* feed there contains at least one route-id with the text 'US'.
- If there are multiple parameters, i.e. the data feed request contains the route-id parameters of: 'US', 'I55', 'OR99' then the Data Portal will validate programmatically that within the *Metadata: Route* metadata feed there contains:
  - At least one route-id with the text 'US'
  - At least one route-id with the text 'I55'
  - At least one route-id with the text 'OR99'

If the inputted parameter value(s) *does not* contain at least one corresponding value stored within the Metadata feed, then the Data Portal Web API will send an HTTP error code "400: Invalid Parameter" response to the external collection process.

- Continuing the example from the scenario above, if the external process's data request contained the route-id parameters of; 'US', 'I55, 'OR99' then the Data Portal will identify that 'I55' is not stored within the *Metadata: Routes* feed. Even though 'US' and 'OR99' are valid parameters, the Data Portal will return only an error code.

If all the inputted parameter value(s) do contain at least one corresponding value stored within the metadata feed, then the service begins the retrieve dataset process.

2.3.2 Dataset Request Parameters

Current datafeed request parameters and definitions can be found listed under each datafeed within the TripCheck API Data Portal: