



For Versions 3.0+

Real-time Data

Request:

Method: HTTP GET

URL: tvhost/realtime_data

Required headers: authorization header as explained below

To get realtime data from all cameras across all connected TrafficVision systems, a client can send a GET request for https://tvhost/realtime_data resource. "tvhost" is the ip address or computer name of the primary (aggregator / TAN) TrafficVision unit.

Response:

Response to the GET for /realtime_data is a JSON object as described below. The order of elements in a JSON object is not guaranteed according to the specs (<http://www.json.org/>). As a result it is highly recommended that a JSON parser should be used to extract the data from the GET response instead of relying on line numbers.

```
{"responses": [{data_1}, {data_2}, ... , {data_N}]}
```

responses is an array of JSON objects, one object for each camera in the system.

Each {data_#} object has following elements relevant to realtime data:

- **timestamp** : (number) unix timestamp of when the response was generated
- **camera_name** : (string) name of the camera as it appears in the TrafficVision system
- **host** : address/hostname of the unit which processes the feed from the camera.
- **camera_index** : (number in range [1...max_num_cameras]) index of the camera relative to the host/box. Each box may have different max_num_cameras. This index is only unique for the individual system and therefore not always unique across all systems.
- **incidents** : (array of objects) list of all current active incidents for the camera. Each object will have the key-values below:
 - **incident_type** : (number)
 - 1: wrong-way vehicle
 - 2: pedestrian in the roadway
 - 3: stopped vehicle
 - 4: congestion
 - 5: slowed traffic
 - 6: low visibility

- **incident_priority** : (number)
 - 1: high
 - 2: medium
 - 3: low
 - **incident_snapshot** : (string) local path to a snapshot/image of the incident.
URL can be constructed as
`"https://tvhost/proxy/" + host + "/" + incident_snapshot`
 - **incident_clip** : (string) local path to a video clip of the incident.
Incident clip recording needs to be enabled in the UI for this feature to work.
URL can be constructed as
`"https://tvhost/proxy/" + host + "/" + incident_clip`
 - **timestamp** : (number) unix timestamp of the incident.
 - **incident_zone** : (string) location of incident on road.
- **calib_data["ptz_mode"]**: (string) used to verify what analytic mode the camera is in.
 - PTZ_APS
 - PTZ_DPA
 - PTZ_MPS
 - PTZ_AUTOLEARN: **realtime_data** will be empty.
 - PTZ_UNKNOWN: **realtime_data** will be empty.
 - **realtime_data** : (object) contains realtime speeds in **lane_mean_speed_data** and counts in **lane_count_data**. *This data is NOT available when **calib_data["ptz_mode"]** is set to PTZ_AUTOLEARN or PTZ_UNKNOWN as shown above.*
 - **inactive** : (number 0 or 1) whether the camera is currently inactive and not collecting data.
 - **inactive_reason** : (string) reason the camera is currently inactive. Possible values are as follows:
 - DISABLED_BY_USER
 - DISABLED_AUTOLEARN_AND_PRESETS
 - AWAY_FROM_PRESET_AND_DISABLED_AUTOLEARN
 - AWAY_FROM_PRESET_AND_LOW_VIEW_QUALITY
 - LOW_FPS
 - **station_id** : (string) optional station_id value, if specified when the camera was added.
 - **latitude** : (number) optional Latitude position of the camera, if specified when the camera was added.
 - **longitude** : (number) optional Longitude position of the camera, if specified when the camera was added.
 - **Additional entries in the object are for internal use only and can be discarded.**

Elements shown in **red** are required as part of other requests described later in the document.

Sample response from a system having only a single processing unit ("**host**": "**127.0.0.1**") and two cameras (camera 'Camera 1' has a congestion incident):

```
{
  "responses": [{
    "timestamp":1494595904,
    "camera_index":1,
    "camera_name":"Camera 1",
    "station_id":"","
    "latitude":0,
    "longitude":0,
    "incidents":[
      {
        "incident_type":4,
```

```

        "incident_priority":2,
        "incident_snapshot":"tmcddata/24/incidents/2017-05-12_09-24-07_incident_congestion_zone_near-lanes.jpg",
        "incident_clip":"tmcddata/24/incidents/2017-05-12_09-24-07_incident_congestion_zone_near-lanes.mp4",
        "timestamp":1494595447,
        "incident_zone":"near-lanes"
    },
    ],
    "inactive":0,
    "inactive_reason":"",
    "calib_data":{
        "ptz_mode":"PTZ_DPA"
    },
    "realtime_data":{
        "lane_mean_speed_data":{
            "lanes_near":2,
            "lanes_far":2,
            "duration_sec":60,
            "timestamp":1494595860,
            "mean_speeds":[
                63.8125,
                72.2142868,
                51.2105255,
                47.7142868
            ]
        },
        "lane_count_data":{
            "lanes_near":2,
            "lanes_far":2,
            "duration_sec":60,
            "timestamp":1494595860,
            "lane_counts_total":[
                16,
                14,
                19,
                7
            ],
            "lane_vph":[
                960,
                840,
                1140,
                420
            ],
            "lane_occupancy":[
                11,
                8,
                16,
                6
            ],
            "lane_density":[
                15,
                11,
                22,
                8
            ]
        },
        "lane_ids":[
            "NS-1",
            "NS-2",
            "FS-1",

```

```

        "FS-2"
    ]
},
"response_id":0,
"host":"127.0.0.1",
"sid":"60e10a5d-1adb-11e7-ae5b-90b11c541a07-24"
},
{
    "timestamp":1494595904,
    "camera_index":2,
    "camera_name":"Camera 2",
    "station_id":"",
    "latitude":0,
    "longitude":0,
    "incidents":[

    ],
    "inactive":0,
    "inactive_reason":"",
    "calib_data":{
        "ptz_mode":"PTZ_AUTOLEARN"
    },
    "realtime_data":{
        "lane_mean_speed_data":{
            "lanes_near":0,
            "lanes_far":0,
            "duration_sec":60,
            "timestamp":1494595860,
            "mean_speeds":[

            ]
        },
        "lane_count_data":{
            "lanes_near":0,
            "lanes_far":0,
            "duration_sec":60,
            "timestamp":1494595860,
            "lane_counts_total":[

            ],
            "lane_vph":[

            ],
            "lane_occupancy":[

            ],
            "lane_density":[

            ]
        },
        "lane_ids":[

        ]
    },
    "response_id":0,
    "host":"127.0.0.1",
    "sid":"60e10a5d-1adb-11e7-ae5b-90b11c541a07-2"
}]
}

```

Clear/Reset incident alert

Request:

Method: HTTP POST

URL: tvhost/proxy/**host**/json_post

(**host** is obtained from the **host** field in the object corresponding to the alert)

Required headers: authorization header similar to that of **/realtime_data**

POST content: JSON formatted object similar to the example below:

```
{
  "request": "cr_reset_incident",
  "request_id": 123,
  "camera_indices": [1],
  "incident_data": {
    "incident_type": 4,
    "clear_reason": "true-create-new-event",
    "clear_note": "Traffic was congested, action required.",
  }
}
```

The values for **camera_indices** (single-element array) and **incident_type** (json-object) can be obtained from the **camera_index** (integer) and **incidents** (json-array of objects) fields of the corresponding object in the **/realtime_data** response.

clear_reason and **clear_note** are both optional. **clear_reason** is an identifier label that can be of your choosing. **clear_note** is a note about the incident. Both of these values will show up in the metadata about the incident.

request_id is a random integer which can be checked later with the **response_id** in the response to determine if the request was processed successfully.

Response:

If the request was processed correctly, the **response_id** will match the **request_id**. It might take between 5-10 seconds for the cleared state of the incident to show up in the polled realtime data.

```
{
  "responses": [
    {
      "response_id": 123,
      "camera_index": 1,
      "sync_flags": {
        "sync_settings": 2,
        "sync_calib": 1,
        "sync_camera_info": 1395842106,
      }
    }
  ]
}
```

```
        "sync_source":1395842106
    }
},
"response_id":123,
"sync_indicator":1395842105
}
```

Authorization Headers

API Key Method:

Clients need to include the following headers in each request for the request to be successful:

- X-TMC-API-USER
- X-TMC-API-KEY

In addition: Content-Type must be “application/json” when sending a POST request to /json_post.