

## Operation Monitoring

Operation Monitoring API refers to any operations dealing with real time information.

- **Vehicle Position**

The operation returns real time vehicle position for given line id's

|                       |  |
|-----------------------|--|
| <b>Endpoint</b>       | <b>https://opendata-api.stib-mivb.be/OperationMonitoring/3.0/VehiclePositionByLine/{line id's}</b> |
| <b>Method</b>         | GET  |
| <b>MimeType</b>       | application/json   |
| <b>Arguments</b>      | a comma separated list of line id's, maximum 10 id's are allowed in the list.                      |
| <b>Returned value</b> | an array of "lines" and for each requested line, an array of "vehiclePositions"                    |

### 1. Sample of a request using Curl

```
> curl -k -X GET --header "Accept: application/json" --header "Authorization: Bearer  
Replace_this_text_by_your_token" "https://opendata-  
api.stibmivb.be/OperationMonitoring/3.0/VehiclePositionByLine/1"
```

### 2. Sample of a request using JQuery

```
var me = this;  
  
// get Vehicle  
Positions $.ajax({  
  url: me.openDataBaseUrl + '/OperationMonitoring/3.0/  
VehiclePositionByLine/' + me.lineId,  
  type: 'GET',  
  error: function (jqXHR, textStatus)  
  { // process error  
  },  
  beforeSend: function setHeader(xhr) {  
    xhr.setRequestHeader('Accept', 'application/json');  
    xhr.setRequestHeader('Authorization', 'Bearer ' + me.apiToken);  
  // add the api token here  
  },  
  success: function (data) {  
    // process the result here  
    console.log('vehiclePosition: ' + data);  
  },  
}); // end of $.ajax({
```

### 3. Sample of the returned value

```
{
  "lines": [
    {
      "lineId": 1,
      "vehiclePositions": [
        {
          "directionId": 8161,
          "distanceFromPoint":0,
          "pointId": 8012
        },
        {
          "directionId": 8731,
```

```
          "distanceFromPoint":0,
          "pointId": 8021
        }
      ]
    }
  ]
}
```

### 4. Remarks:

- a. Note that the position of a vehicle is given in the following format:
  - "directionId": the direction of the vehicle as the terminal "pointId"
  - "distanceFromPoint": the distance (in meters) covered by a vehicle since the last point represented by the "pointId" attribute
  - "pointId": the last Stop crossed by a vehicle
- b. *Note about the Id's usage:*
  - The Line Id used as argument of the operation refers to the field "route\_short\_name" of the GTFS file "routes.txt"
  - The returned "directionId" and "pointId" refer to the fields "stop\_id" of the GTFS file "stops.txt"
- c. *Lifetime:*
  - A vehicle position has a lifetime of about 20 seconds, therefore it is not necessary to poll the service at a higher frequency.
  - The array of vehiclePositions may include some technical stops. These kind of stops are not published in the GTFS stops files, which is why you should ignore them.