Operation Monitoring

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

- **Waiting Time**

The operation returns the waiting times for the next two vehicles of each line passing through the requested stop id's.

Endpoint	**https://opendata-api.stib-mivb.be/OperationMonitoring/4.0/PassingTimeByPoint/{point id's}**
Method	GET
MimeType	application/json
Arguments	a comma separated list of Point Id's (a.k.a. Stop Id's), maximum 10 id's are allowed in the list
Returned value	an array of "points" and for each requested point, an array of "passingTimes"

1. ***Sample of a request using Curl***

   ➢ curl -k -X GET --header "Accept: application/json" --header "Authorization: Bearer 30ca85ad55a0e384772b653e149e39a5" https://opendata-api.stib-mivb.be/OperationMonitoring/4.0/PassingTimeByPoint/8031

2. ***Sample of a request using JQuery***

```
        url: me.openDataBaseUrl + '/OperationMonitoring/4.0/
    PassingTimeByPoint/' + item.join("%2C"), // item is an array of
    id's, joined to get a comma separated list
        type: 'GET',
        error: function (jqXHR, textStatus)
            { // process error
        },
        beforeSend: function setHeader(xhr) { xhr.setRequestHeader('Accept',
            'application/json'); xhr.setRequestHeader('Authorization', 'Bearer
            ' + me.apiToken);
        },
        success: function (data) {
            // process the result here
            console.log('PassingTimeByPoint: ' + data);
        },
     })// end of $.ajax({
```

3. ***Sample of the returned value***


```
{
  "points": [
    {
      "passingTimes": [
        {
          "destination": {
            "fr": "GARE DE L'OUEST",
            "nl": "WESTSTATION"
          },
          "expectedArrivalTime": "2018-06-26T11:57:00+02:00",
          "lineId": "1"
        },
        {
          "destination": {
            "fr": "ERASME",
            "nl": "ERASMUS"
          },
          "expectedArrivalTime": "2018-06-26T11:56:00+02:00",
          "lineId": "5"
        }

      ],
      "pointId": "8011"
    },
    {
      "passingTimes": [
        {
          "destination": {
            "fr": "UZ BRUSSEL",
            "nl": "UZ BRUSSEL"
          },
          "lineId": "13",
          "message": {
            "fr": "LIGNE DÉVIÉE",
            "nl": "LIJN OMGELEID"
          }
        } ,
        {
          "destination": {
            "fr": "UZ BRUSSEL",
            "nl": "UZ BRUSSEL"
          },
          "lineId": "15",
          "message": {
            "fr": "LIGNE DÉVIÉE",
            "nl": "LIJN OMGELEID"
          }
        }
      ],
      "pointId": "1000"
    }
  ]
}
```

4. *Remarks:*

a. Note about the Id's usage:

➔ The Point id used as argument of the operation refers to the fields "stop_id" of the GTFS file "stops.txt"

➔ The returned "lineId" refers to the field "route_short_name" of the GTFS file "routes.txt".
➔ If the vehicle will not pass by the stop, the API doesn't return its waiting time.
➔ An information message might be included in the result with one of the following values :
- Line diverted
- Last departure
- Theoretical time
- Blocked Vehicle
- End of service
- Consult timetable
- Don't embark

b. Lifetime:
A vehicle position has a lifetime of about 20 seconds, therefore it is not necessary to poll the service at a higher frequency.