Live Push API

| Live Push API V0.9 | |
|---|---|
| Date | 21st of March 2017 |
| Update | 4th of October 2023 |

**Version History**

| Date | Version | Author | Comments |
|---|---|---|---|
| 21-03-2017 | V.0.1 | ED | Initial version |
| 24-03-2017 | V.0.2 | MJ | Fix some textual errors |
| 25-04-2017 | V0.3 | MB | Parameter table added |
| 12-04-2021 | V0.4 | SN | Use same font in whole parameter table.<br>Correct information about accuracy field. |
| 20-04-2021 | V0.5 | TvD | Push API secret explanation added |
| 19-01-2022 | V0.6 | SN | Add iginition field |
| 09-06-2022 | V0.7 | SN | Add (Java) models |
| 03-03-2023 | V0.8 | SN | Push API secret token vs key confusion corrected.<br>Add information about objects pushed. |
| 04-10-2023 | V0.9 | MJ | Add supported hardware types |

### 1) Intro

The Live API is the gate to achieve near real-time points from Moving Intelligence products.
It can be used to integrate Moving Intelligence products in your own platform.
By integrating it, you can also customize it further or enrich it with new data.
To be able to use the API you must have a Moving Intelligence account with registered devices and API push settings properly set.

### 2) Settings

The push API settings can be found in **CONFIGURATION → PUSH API**, on this page you can edit the connection URL and set an optional API secret.
This secret, when set, will be included with every push in the X-API-TOKEN header.

The connection will be tested after saving it, the response will show in the status field.
Please note the payload for this test call is an empty string ("").

When the response from the endpoint is HTTP 200 (OK), the URL will be saved and used for sending positions.
All other response codes will cause the old URL to be used (or no positions will be sent if no URL was present).

**3) Info**

Live API collects and pushes every minute new data received from the registered devices in your account.
This will include all devices supported in the Push API within your account / contractor.

Supported hardware types are Mi04, Mi50 and Mi01 Tracer.

The push message contains an array of LivePositionWrapper objects.

The LivePositionWrapper object contains the link to the entity/object named as entityId.
This field can be found in the moving intelligence platform in **MANAGE** → **OBJECTS** as field **ID**.

These two fields are the same and can be matched to find out to which entity/object this entityId belongs.
More information about the push data is noted in the next section.

**4) PUSH DATA**

The structure of LivePositionWrapper object is described below.

| **LivePositionWrapper** | | |
|---|---|---|
| **Parameter** | **Type** | **Description** |
| entityId | Long | Identifier for the device |
| livePositions | Array of objects | Data wrapper which contains an array of position information |
| **LivePositions** | | |
| **Parameter** | **Type** | **Description** |
| createDate | Long | Date in epoch format in seconds, event creation moment on the device |
| gpsDate | Long | Epoch GPS date in seconds at event creation moment (only available for GPS based positions). |
| valid | Boolean | If the GPS position was based on valid data (only available for GPS based positions). |
| recent | Boolean | If the GPS position was based on recent data (only available for GPS based positions). |
| latitude | Double | Angular distance of any object from the equator measured in degrees. |
| longitude | Double | Angular distance from the zero meridian reference measured in degrees. |
| speed | Integer | Speed in kilometers per hour (only available for GPS based positions). |
| direction | Integer | Direction in degrees (only available for GPS based positions). |
| accuracy | Integer | Accuracy of this position in meters (is 0 on GPS based positions and > 0 when position is based on GSM triangulation). |
| ignition | Boolean | Ignition state. True means ignition is on and false means ignition is off. Null when ignition state is unknown or not supported. |

**Model classes (Java)**

```java
public class LivePositionWrapper {

    /**
     * Device identifier.
     */
    private Long               entityId;
    /**
     * List of position for the device, currently only holds one position (the most recent).
     */
```

```
    private List<LivePosition> livePositions;
}

public class LivePosition {

    /**
     * Epoch date this point was created on the device.
     */
    private Long    createDate;
    /**
     * Epoch date of the GPS.
     * Note: Only available for GPS based positions.
     */
    private Long    gpsDate;
    /**
     * Was the data in this point based on a valid position?
     * Note: Only available for GPS based positions.
     */
    private Boolean valid;
    /**
     * Was the data in this point recent?
     * Note: Only available for GPS based positions.
     */
    private Boolean recent;
    /**
     * Latitude.
     */
    private Double  latitude;
    /**
     * Longitude.
     */
    private Double  longitude;
    /**
     * Speed in kilometers per hour.
     * Note: Only available for GPS based positions.
     */
    private Integer speed;
    /**
     * Direction in degrees.
     * Note: Only available for GPS based positions.
     */
    private Integer direction;
    /**
     * Accuracy of this position in meters.
     * Note: When 0 this position is GPS based, > 0 means GSM-cell triangulation based.
     */
    private Integer accuracy;
    /**
     * State of ignition (+15).
     * When null, ignition state could not be determined (yet).
     * Note: Only available for devices with connection to ignition wire.
     */
    private Boolean ignition;
}
```

**JSON example**

```
[{
    "entityId": 29776,
    "livePositions": [{
        "createDate": 1489995390,
        "gpsDate": 1489992748,
        "valid": true,
        "recent": true,
        "latitude": 52.432855,
        "longitude": 5.420902,
        "speed": 99,
        "direction": 226,
        "accuracy": 0,
        "ignition": true
    }]
}, .....];
```