



Universidade do Minho
Departamento de Informática

SENSORIZAÇÃO E AMBIENTE

MESTRADO EM ENGENHARIA INFORMÁTICA, 1º ANO - Perfil SI



Universidade do Minho
Departamento de Informática



Soft Sensors



- Soft Sensors
 - Virtual Sensors
- Hands On





Universidade do Minho
Departamento de Informática

Soft Sensors

Virtual Sensors



Virtual Sensors

- Virtual sensors are **algorithms or software-based models** that estimate or infer measurements that would typically require physical sensors
- Instead of directly measuring a physical quantity, virtual sensors use **mathematical models, machine learning, or sensor fusion techniques** to generate an estimated value based on available data from other sensors or sources
- Virtual sensors pipeline:
 - Data Collection – gathering data from multiple physical sensors, databases, or simulations
 - Data Processing – applying algorithms, AI models, or statistical techniques to estimate the desired variable
 - Estimation Output – providing a computed or inferred value that can be used in decision-making, monitoring, or automation
- Areas of application:
 - Automotive
 - Healthcare
 - Manufacturing & Industry 4.0
 - Smart Cities
 - Aerospace



Virtual Sensors

How to use virtual sensors:

- make **API calls** through **HTTP Get Requests**
- **parse** the received **JSON** (or XML)
- (optional) use an **API key**
- There is a limited number of requests to make



Virtual Sensors

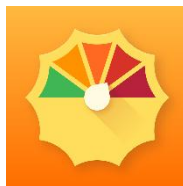
For **Climatological Data**:





Virtual Sensors

For **Air Pollution Data**:



OpenUV



AQICN

<https://openweathermap.org/api>

<https://www.openuv.io/api>

<https://www.iqair.com/commercial-air-quality-monitors/api>

<https://docs.openaq.org/>

<https://docs.breezometer.com/>

<https://aqicn.org/json-api/doc/>



Open Weather



Weather in your city

[Guide](#)

[API](#)

[Dashboard](#)

[Marketplace](#)

[Pricing](#)

[Maps](#)

[Our Initiatives](#)

[Partners](#)

[Blog](#)

[For Business](#)

[Sign in](#)

[Support centre](#)

Weather API

[Home](#) / [Weather API](#)

Please, [sign up](#) to use our fast and easy-to-work weather APIs. As a start to use OpenWeather products, we recommend our [One Call API 3.0](#). For more functionality, please consider our products, which are included in [professional collections](#).

One Call API 3.0

[API doc](#)

[Subscribe](#)

Pay as you call

Make an API call to receive access to the various data:

- **Current weather and forecasts:**
 - minute forecast for 1 hour
 - hourly forecast for 48 hours
 - daily forecast for 8 daysand government weather alerts
- **Weather data for any timestamp** for 46+ years historical archive and 4 days ahead forecast
- **Daily aggregation** of weather data for 46+ years archive and 1.5 years ahead forecast
- **Weather overview** with a human-readable weather summary for today and tomorrow's forecast

1,000 API calls per day for free
0.0014 EUR per API call over the daily limit

[Subscribe](#)

This is a separate subscription plan, which includes only One Call API.

Read more about this API and subscription plan in the [FAQ](#).

Professional collections

For professionals and specialists with middle sized project, we recommend our Professional collections, which included [Current & Forecasts collection](#), [Historical weather data collection](#), [Weather Maps collection](#) and other APIs.

For Enterprise level projects we provide Enterprise license, which is included all forecast products and current state, along with alerts, maps, and other products. [Learn more](#)

You can read the [How to Start](#) guide and enjoy using our powerful weather APIs right now.



Ulla Weather Assistant



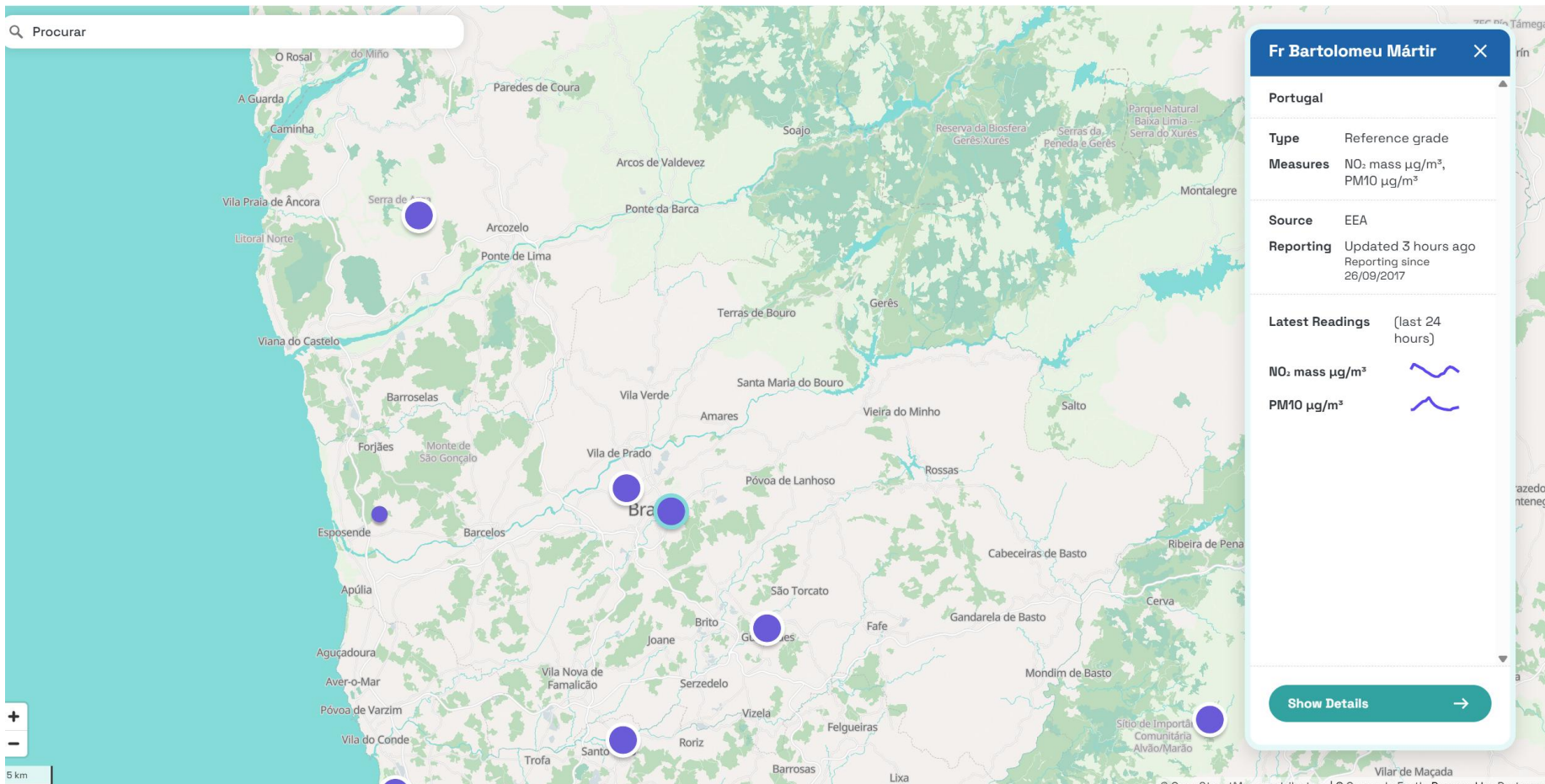
Open Weather

`https://api.openweathermap.org/data/3.0/weather?lat=41.56&lon=-8.40&units=metric&lang=pt&appid={API key}`

```
{"coord":{"lon":-8.4,"lat":41.56},"weather":[{"id":804,"main":"Clouds","description":"nublado","icon":"04d"}],"base":"stations","main":{"temp":13.98,"feels_like":13.21,"temp_min":13.98,"temp_max":13.98,"pressure":1004,"humidity":68,"sea_level":1004,"grnd_level":975},"visibility":10000,"wind":{"speed":2.38,"deg":249,"gust":4.1},"clouds":{"all":98},"dt":1741795731,"sys":{"type":2,"id":2021760,"country":"PT","sunrise":1741762228,"sunset":1741804581},"timezone":0,"id":2734134,"name":"Senhora da Rocha","cod":200}
```



Open Air Quality

[Explore the data](#)[Why air quality?](#)[Why open data?](#)[Partners](#)[Developers](#)[About](#)[Sign up](#)[Login](#)[Donate](#)



Open Air Quality

```
curl --request GET \ --url "https://api.openaq.org/v3/locations/8118" \ --header "X-API-Key:  
API Key"
```

```
{"meta":{"name":"openaq-api","website":"/","page":1,"limit":100,"found":1},"results":[{"id":8118,"name":"New Delhi","locality":"India","timezone":"Asia/Kolkata","country":{"id":9,"code":"IN","name":"India"},"owner":{"id":4,"name":"Unknown Governmental Organization"},"provider":{"id":119,"name":"AirNow"},"isMobile":false,"isMonitor":true,"instruments":[{"id":2,"name":"Government Monitor"}],"sensors":[{"id":23534,"name":"pm25 µg/m³","parameter":{"id":2,"name":"pm25","units":"µg/m³","displayName":"PM2.5"}}],"coordinates":{"latitude":28.63576,"longitude":77.22445},"licenses":[{"id":33,"name":"US Public Domain","attribution":{"name":"Unknown Governmental Organization","url":null},"dateFrom":"2016-01-30","dateTo":null}],"bounds":[77.22445,28.63576,77.22445,28.63576],"distance":null,"datetimeFirst":{"utc":"2016-11-09T19:00:00Z","local":"2016-11-10T00:30:00+05:30"},"datetimeLast":{"utc":"2025-03-12T14:30:00Z","local":"2025-03-12T20:00:00+05:30"}}
```



Open UV



OpenUV.io

API Limit:

50 Reqs/Day



Get UV Index

GET /api/uv

Get UV Index Forecast

GET /api/forecast

Get Requests Stat

GET /api/stat

Check API Status

GET /api/status

FAQ

And Knowledge Base

API Key

openuv-c10rrm864ko56-io



COPY

ⓘ Add this API Key to x-access-token header of your request. See [code](#) samples below

Get Real-Time UV Index data

Get real-time UV Index data

Endpoint

GET <https://api.openuv.io/api/v1/uv?lat=:lat&lng=:lng&alt=:alt&dt=:dt>

Parameters

Latitude, from -90.00 to 90.00:

51.5

Longitude, from -180.00 to 180.00:

-0.11

ⓘ Drag Google Maps Marker to set your location



Billing & Invoices

● UNLOCK START-UP PLAN

for \$15 USD/Month

Powered by stripe

ⓘ Start-Up plan includes up to 15000 Requests per day. We never store your billing details on our servers and use **Stripe** for billing.

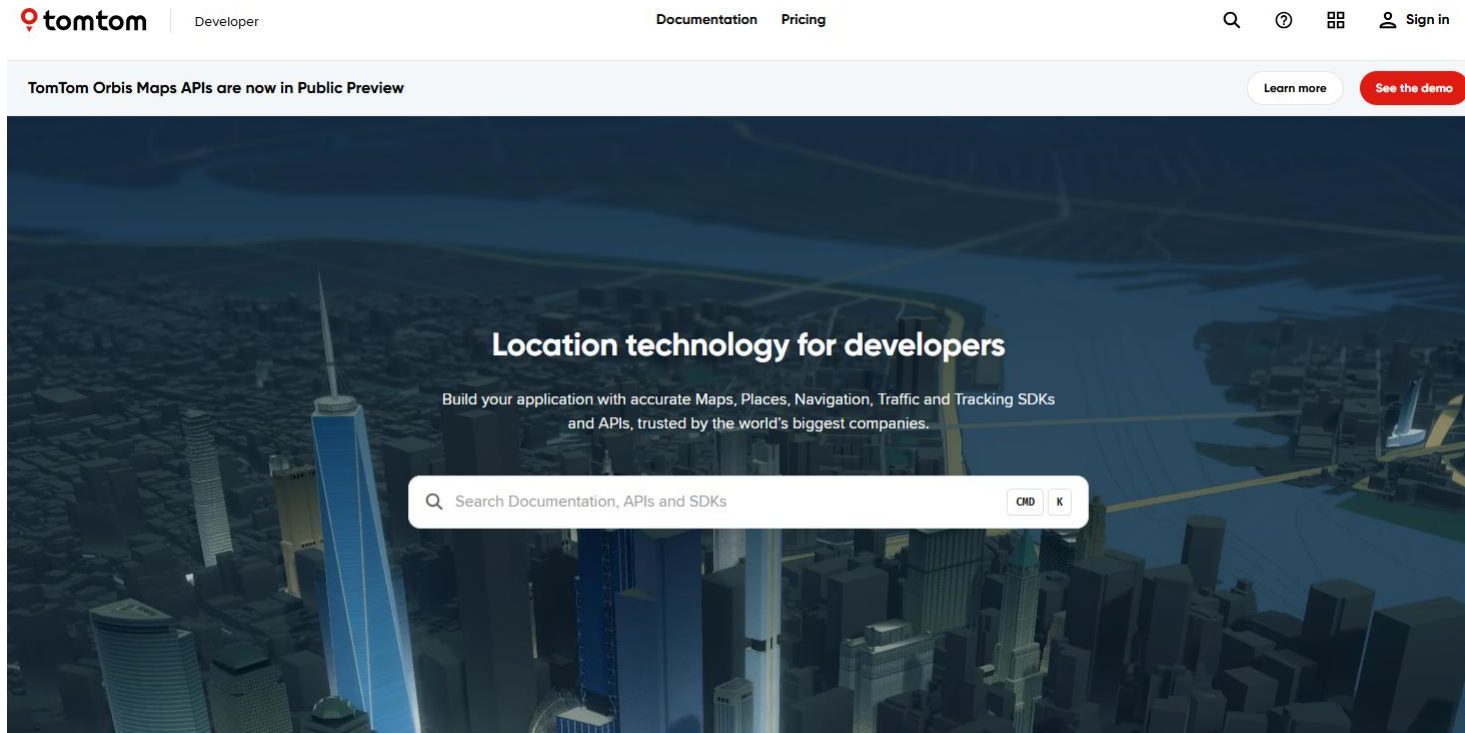
JSON Response

```
{ 1 item
  "result": { 8 items
    "uv": 0.4055
    "uv_time": "2025-03-12T16:19:11.434Z"
    "uv_max": 2.7168
    "uv_max_time": "2025-03-12T12:11:35.302Z"
    "ozone": 332.5
    "ozone_time": "2023-04-12T15:04:31.773Z"
    "safe_exposure_time": { 6 items
      "st1": 411
      "st2": 493
      "st3": 658
      "st4": 822
      "st5": 1315
      "st6": 2466
    }
  }
  "sun_info": { 2 items
    "sun_times": { 14 items
      "solarNoon": "2025-03-12T12:11:35.302Z"
      "nadir": "2025-03-12T06:11:35.302Z"
      "sunrise": "2025-03-12T06:22:43.639Z"
      "sunset": "2025-03-12T18:00:26.964Z"
      "sunriseEnd": "2025-03-12T06:26:09.780Z"
      "sunsetStart": "2025-03-12T17:57:00.823Z"
    }
  }
}
```



Virtual Sensors

- For **Traffic Flow Data**:



SDKs

Build a complete maps, navigation or automotive experience with minimal





TomTom's Traffic API

https://api.tomtom.com/map/1/tile/basic/main/0/0/0.png?view=Unified&key=YOUR_API_KEY





TomTom's Traffic API

Traffic API

Documentation

API Explorer

TomTom Maps

Product Information

Introduction

Release notes

Market Coverage

Region-specific content

Traffic Incidents

Traffic Flow

Map Styles

Vector Tile Structure

Introduction

Service version: 4 & 5

Last edit: 2024.08.09

What is TomTom's Traffic API?

The Traffic API is a suite of web services designed for developers to create web and mobile applications around real-time traffic. These web services can be used via RESTful APIs. The TomTom Traffic team offers a wide range of solutions to enable you to get the most out of your applications. Make use of the real-time traffic products or the historical traffic analytics to create applications and analysis that fits the needs of your end-users.

The offering is split into two services with their respective endpoints:

- **Traffic Incidents:** This provides an accurate view about *traffic jams* and *incidents* around a road network.
- **Traffic Flow:** This provides *real time observed speeds* and *travel times* for all key roads in a network.

The Traffic API web services are based on the real-time traffic data TomTom Traffic™, and consists of the the following services with endpoints:

Traffic Incidents

The Traffic Incidents service is a suite of web services that provides an accurate view about traffic jams and incidents around a road network.

Traffic Flow

The Traffic Flow service is a suite of web services that provides real time observed speeds and travel times for all key roads in a network.

On this page

What is TomTom's Traffic API?

Getting started

Features

TomTom's Traffic Incident service

TomTom's Traffic Flow service

Common use cases you can implement

Display

Indicate

Combine



TomTom's Traffic Flow API

Traffic API

Documentation

API Explorer

Product Information ▾

Traffic Incidents ▾

Traffic Flow ▾

Traffic Flow service

Flow Segment Data GET

Raster Flow Tiles GET

Vector Flow Tiles GET

Map Styles ▾

Vector Tile Structure ▾

Flow Segment Data

Service version: 4

Last edit: 2022.08.15

Purpose

This service provides information about the speeds and travel times of the road fragment closest to the given coordinates. It is designed to work alongside the Flow Tiles to support clickable flow data visualizations. With this API, the client side can connect any place in the map with flow data on the closest road and present it to the user.

Run this endpoint

You can easily run this and other endpoints. Go to the TomTom [API Explorer](#) page and follow the directions.

Request data

HTTPS method: GET

- Constants and parameters enclosed in curly brackets `{ }` must be replaced with their values.
- Please see the following [Request parameters](#) section with the required and optional parameters tables for these values. The generic URL format is as follows.

On this page

Purpose

Run this endpoint

Request data

HTTPS method: GET

Request parameters

Request headers

Response data

Successful response

Error response

Response codes

Response headers



TomTom's Traffic Flow API

https://api.tomtom.com/traffic/services/4/flowSegmentData/absolute/10/xml?key={Your_API_Key}&point=41.56171,-8.39729

```
▼<flowSegmentData version="traffic-service-flow 1.0.133">
  <frc>FRC2</frc>
  <currentSpeed>26</currentSpeed>
  <freeFlowSpeed>42</freeFlowSpeed>
  <currentTravelTime>65</currentTravelTime>
  <freeFlowTravelTime>40</freeFlowTravelTime>
  <confidence>1.000000</confidence>
  <roadClosure>false</roadClosure>
  ▼<coordinates>
    ▼<coordinate>
      <latitude>41.560614</latitude>
      <longitude>-8.393874</longitude>
    </coordinate>
    ▼<coordinate>
      <latitude>41.560500</latitude>
      <longitude>-8.394084</longitude>
    </coordinate>
    ▼<coordinate>
      <latitude>41.560430</latitude>
      <longitude>-8.394208</longitude>
    </coordinate>
    ▼<coordinate>
      <latitude>41.560339</latitude>
      <longitude>-8.394366</longitude>
    </coordinate>
    ▼<coordinate>
      <latitude>41.560259</latitude>
      <longitude>-8.394507</longitude>
    </coordinate>
    ▼<coordinate>
      <latitude>41.560130</latitude>
      <longitude>-8.394731</longitude>
    </coordinate>
    ▼<coordinate>
      <latitude>41.560000</latitude>
      <longitude>-8.394854</longitude>
    </coordinate>
  </coordinates>
</flowSegmentData>
```



TomTom's Traffic Incidents API

Traffic API

Documentation

API Explorer

Product Information ▾

Traffic Incidents ▾

Traffic Incidents service

Traffic Model ID

Incident Details GET/POST

Incident Viewport GET

Raster Incident Tiles GET

Vector Incident Tiles GET

Incident Details (Deprecated) GET

Traffic Flow ▾

Map Styles ▾

Vector Tile Structure ▾

Incident Details

Service version: 5

Last edit: 2024.11.05

Important note

This is the new and improved version (5) of the Incident Details endpoint. We highly recommend that you use this new version 5 rather than the old version 4, which we marked as deprecated.

Purpose

The Incident Details service provides information on traffic incidents which are inside a given bounding box or whose geometry intersects with it. The freshness of data is based on the provided Traffic Model ID (**t**). The data obtained from this service can be used as standalone or as an extension to other Traffic Incident services.

In the second case, the same Traffic Model ID should be used when calling all services in order to grant synchronization of data between APIs. Apart from present incident data, it is possible to get information about planned future incidents.

Run this endpoint

You can easily run this and other endpoints. Go to the TomTom [API Explorer](#) page and follow the directions.

On this page

Important note

Purpose

Run this endpoint

Request data

HTTPS method: GET

HTTPS method POST

Request parameters

POST request body

Request headers

Response data

Successful response

Error response

HTTP response codes

HTTP response headers



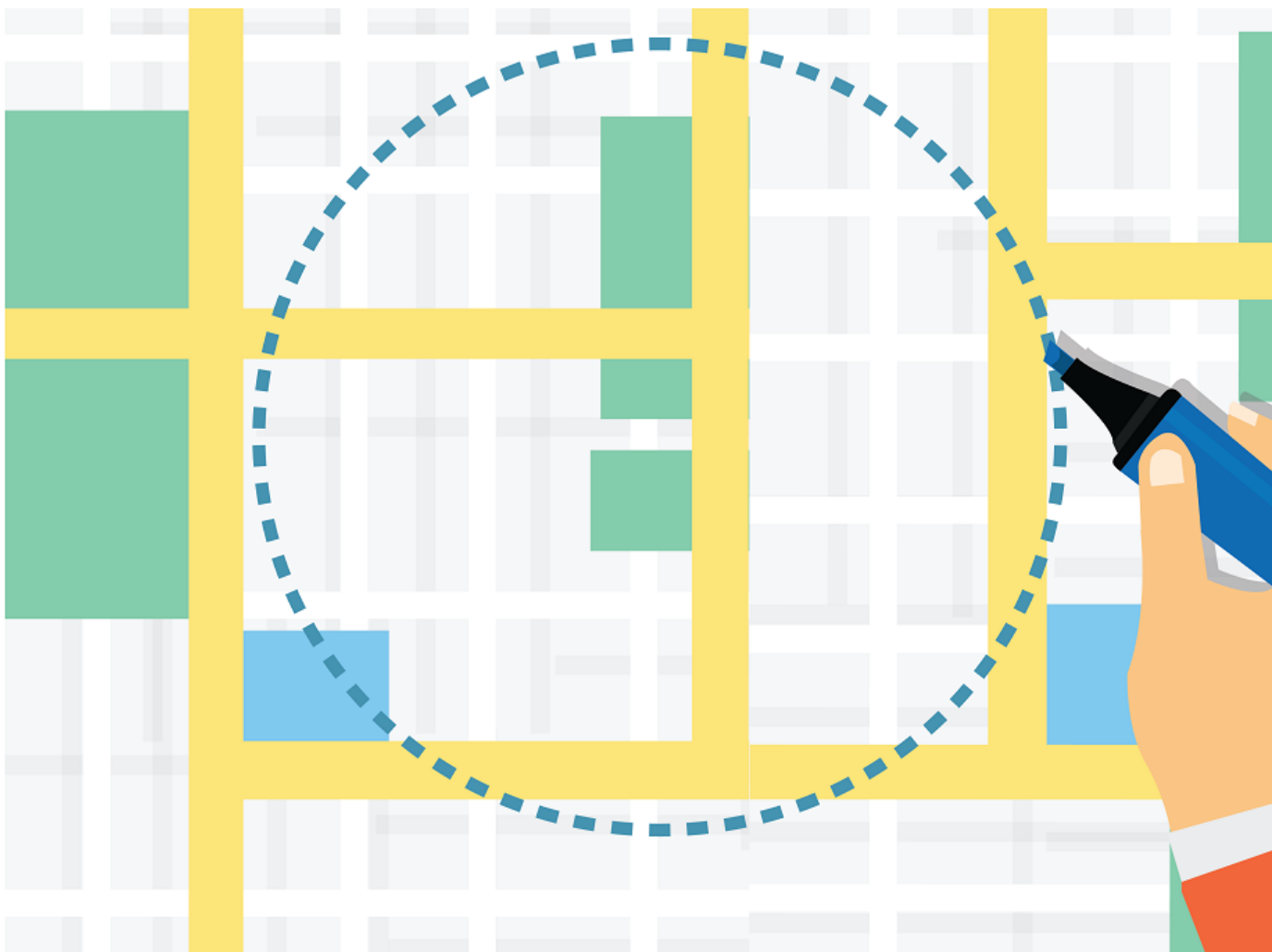
TomTom's Traffic Incidents API

`https://api.tomtom.com/traffic/services/5/incidentDetails?key={Your_Api_Key}&ids=4819f7d0a15db3d9b0c3cd9203be7ba5&fields={incidents{type,geometry{type,coordinates}},properties{iconCategory}}&language=pt-PT&t=1111&timeValidityFilter=present`

```
{"incidents":[null]}
```



Geofencing





Google's Geofencing API

- A **location aware service** that establishes **virtual perimeters** for **real-world geographic areas**
- Vulnerable Road Users are warned as soon as they enter/exit a geofence
- Notifications about **dangerous places, polluted zones, road works, concerts, events** and others

✗ First try:



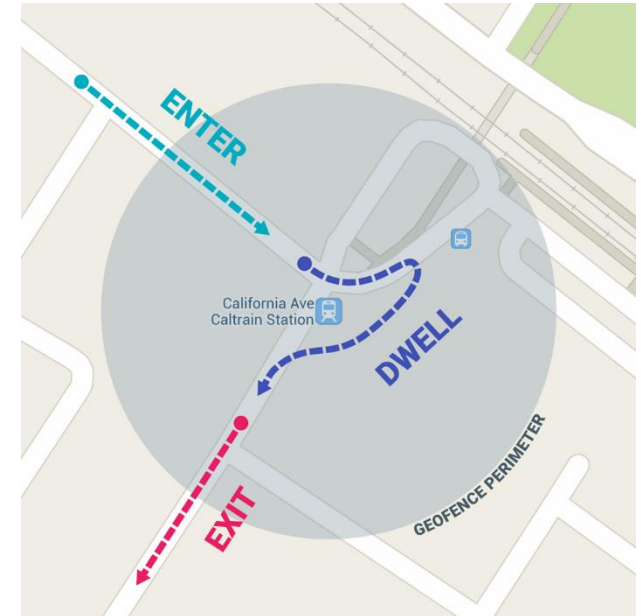
✓ Second try:





Google's Geofencing API

- A **geofence** is defined by a **latitude**, a **longitude** and a **radius**
- With this API it is possible to have 100 active geofences per app
- For each geofence, Location Services can be asked to send **entrance** and **exit events**
- The **limit the duration of any geofence** can be set by specifying an expiration duration in milliseconds. As soon it expires, it is removed





Geofencing Monitoring

- Create an instance of the geofencing client

```
public class MainActivity extends AppCompatActivity {  
  
    // ...  
    private GeofencingClient geofencingClient;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
  
        //...  
  
        geofencingClient = LocationServices.getGeofencingClient(this);  
    }  
}
```




Geofencing Monitoring

- Add the following permissions as child elements of the `<manifest>` element in your app's manifest

```
<uses-permission android:name = "android.permission.ACCESS_FINE_LOCATION" />
```

```
<!-- Required if your app targets Android 10 (API level 29) or higher -->
```

```
<uses-permission android:name = "android.permission.ACCESS_BACKGROUND_LOCATION" />
```

- Use a `BroadcastReceiver` to listen for geofence transitions. Add it as a child of the `<application>` element

```
<application  
    android:allowBackup = "true" >  
    ...  
    <receiver android:name = ".GeofenceBroadcastReceiver" />  
</application>
```



Geofencing Monitoring

- Create an instance of the geofencing client

```
public class MainActivity extends AppCompatActivity {  
  
    // ...  
    private GeofencingClient geofencingClient;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
  
        //...  
  
        geofencingClient = LocationServices.getGeofencingClient(this);  
    }  
}
```



Geofencing Monitoring

- Create and add geofences by creating geofence objects (it can be in the same activity as previously)

```
ArrayList<Geofence> geofenceList = new ArrayList<String>();
geofenceList.add(new Geofence.Builder()
    //request ID of the geofence, i.e., a string to identify the geofence
    .setRequestId(geo_name)
    .setCircularRegion(
        latitude,
        longitude,
        radius
    )
    .setExpirationDuration(Geofence.NEVER_EXPIRE)
    .setTransitionTypes(Geofence.GEOFENCE_TRANSITION_ENTER |
        Geofence.GEOFENCE_TRANSITION_EXIT)
    .build()
);
```

- The `GEOFENCE_TRANSITION_ENTER` transition triggers when a device enters a geofence, and the `GEOFENCE_TRANSITION_EXIT` transition triggers when a device exits a geofence.



Geofencing Monitoring

- Use the `GeofencingRequest` class and its nested `GeofencingRequest.Builder` to specify the geofences to monitor and to set how related geofence events are triggered (it can be in the same activity as previously)

```
private GeofencingRequest getGeofencingRequest() {  
    GeofencingRequest.Builder builder = new GeofencingRequest.Builder();  
    builder.setInitialTrigger(GeofencingRequest.INITIAL_TRIGGER_ENTER);  
    builder.addGeofences(geofenceList); //list populated in the previous slide  
    return builder.build();  
}
```

- Specifying `INITIAL_TRIGGER_ENTER` tells that `GEOFENCE_TRANSITION_ENTER` should be triggered if the device is already inside the geofence



Geofencing Transitions

- The `BroadcastReceiver` created previously will be used to handle a geofence transition (because it will be listening, and it will only update when an event occurs - transition into or out of a geofence). A `PendingIntent` can be defined to start a `BroadcastReceiver` (it can be in the same activity as previously)

```
public class MainActivity extends AppCompatActivity {  
    // ...  
    private PendingIntent getGeofencePendingIntent() {  
        //reuse the PendingIntent if we already have it  
        if (geofencePendingIntent != null) {  
            return geofencePendingIntent;  
        }  
        Intent intent = new Intent(this, GeofenceBroadcastReceiver.class);  
        //FLAG_UPDATE_CURRENT to get the same pending intent  
        geofencePendingIntent = PendingIntent.getBroadcast(this, 0, intent, PendingIntent.  
            FLAG_UPDATE_CURRENT);  
        return geofencePendingIntent;  
    }  
}
```



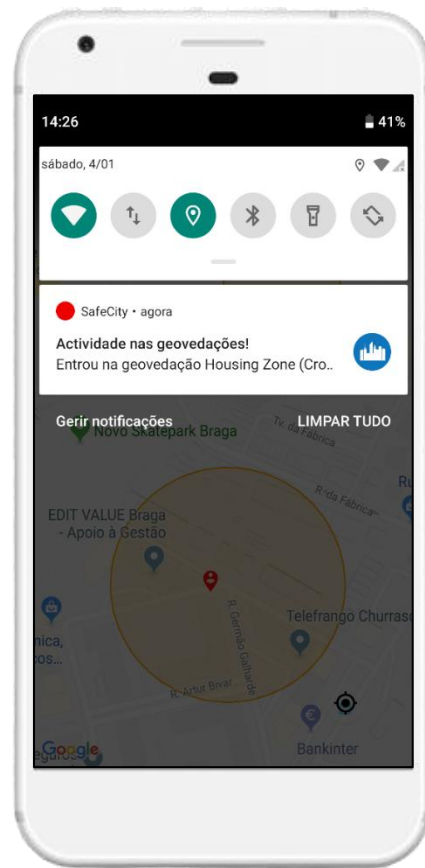
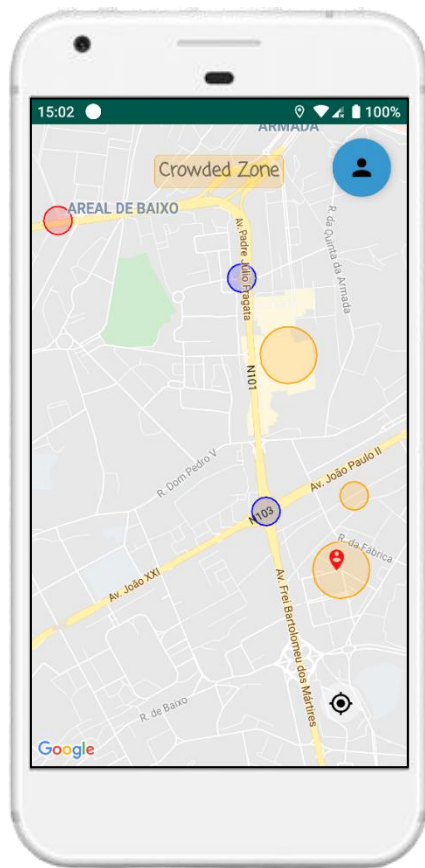
Geofencing Transitions

- Now add the geofences to the `geofencingClient` object created previously. It should be provided as arguments the `GeofencingRequest` object (`getGeofencingRequest()`) and the `PendingIntent` (`getGeofencePendingIntent()`)

```
geofencingClient.addGeofences(getGeofencingRequest(), getGeofencePendingIntent())
    .addOnSuccessListener(this, new OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {
            //geofences added
        }
    })
    .addOnFailureListener(this, new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            //failed to add geofences
        }
    });
```



Google's Geofencing API





Hands On

Discover and implement:

- **For even student numbers**

- Design a data collector (in any programming language) capable of periodically collecting **climatological data** from **OpenWeather** and **OpenAQ**. Save the captured data in an Adafruit or Firebase feed

- **For odd student numbers**

- Design a data collector (in any programming language) capable of periodically collecting **road traffic flow data** from **TomTom's Traffic Flow API**. Save the captured data in an Adafruit or Firebase feed

- **Extra**

- Implement a set of **geofences** using **Google's Geofencing API** and display these geofences in a mobile and/or web application. The characteristics of the geofences, i.e. latitude, longitude and radius, can be defined in an Adafruit or Firebase feed