



# **Vélo Epicurien**

**GLO-4035**

Guillaume HECTOR  
Jérémy LEVILAIN  
Jonathan BIBAS

# Sommaire

<b>Sommaire</b>	<b>2</b>
<b>Introduction</b>	<b>3</b>
<b>Stratégie d'acquisition des données</b>	<b>4</b>
Itinéraires	4
Restaurants	4
<b>Technologies utilisées</b>	<b>6</b>
Langage de programmation	6
Base de données	6

# Introduction

Ce document vise à décrire le plus précisément possible la direction technique de notre projet afin de prouver sa faisabilité. Il vise aussi à expliquer à la fois notre stratégie d'acquisition des données ainsi que nos choix de technologies au sein du projet.

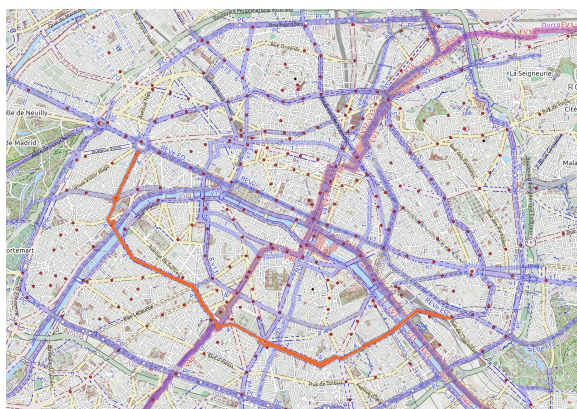
# Stratégie d'acquisition des données

Nous avons choisi Paris comme ville ciblée par notre projet de Vélo Épicurien.

## Itinéraires

La récolte des données cartographiques ainsi que les données concernant les routes et accès pour les vélos qui sera tirée des informations disponibles sur OpenStreetMap via leur API ouverte, gratuite et communautaire. Son approche communautaire permet d'avoir des résultats très précis, les contributeurs étant généralement plus motivés et précis que des entreprises privées.

Dans l'exemple ci-dessous, un itinéraire en vélo calculé sur l'interface d'OpenStreetMap et sa correspondance JSON via l'API :



[Lien](#)

```
{
  "version": "0.6",
  "generator": "CGImap 0.8.3 (2064127 spike-07.openstreetmap.org)",
  "copyright": "OpenStreetMap and contributors",
  "attribution": "http://www.openstreetmap.org/copyright",
  "license": "http://opendatacommons.org/licenses/odbl/1-0/",
  "elements": [
    {
      "type": "node",
      "id": 361730,
      "lat": 48.8311993,
      "lon": 2.3543524,
      "timestamp": "2018-10-19T08:14:28Z",
      "version": 8,
      "changeset": 63670386,
      "user": "Chlc",
      "uid": 2322305,
      "tags": {
        "highway": "traffic_signals"
      }
    },
  ],
}
```

[Lien](#)

Comme on peut le voir sur la représentation JSON, l'itinéraire est une succession d'éléments représentant le trajet : signalétique, angles, croisements, routes, etc. Nous utiliserons ce mécanisme d'enchaînement d'éléments afin d'optimiser l'ingestion des données.

## Restaurants

La récolte des données des restaurants se fera via l'API de Google "Place Search" afin de trouver des restaurants proches, on utilisera en plus pour compléter les données l'API de TheFork.com.

Les données seront reçues sous forme de JSON dont voilà un exemple pour un musée :

```

{
  "candidates" : [
    {
      "formatted_address" : "140 George St, The Rocks NSW 2000, Australia",
      "geometry" : {
        "location" : {
          "lat" : -33.8599358,
          "lng" : 151.2090295
        },
        "viewport" : {
          "northeast" : {
            "lat" : -33.85824767010727,
            "lng" : 151.2102470798928
          },
          "southwest" : {
            "lat" : -33.86094732989272,
            "lng" : 151.2075474201073
          }
        }
      },
      "name" : "Museum of Contemporary Art Australia",
      "opening_hours" : {
        "open_now" : false,
        "weekday_text" : []
      },
      "photos" : [
        {
          "height" : 2268,
          "html_attributions" : [
            "\u003ca href=\"https://maps.google.com/maps/contrib/113202928073475129698/\">
          ],
          "photo_reference" : "CmRaAAAAfxSORBfVmhZcERd-9eC5X1x1pKQgbmunjoYdGp4dYADIqC0AX",
          "width" : 4032
        }
      ],
      "rating" : 4.3
    }
  ],
  "debug_log" : {
    "line" : []
  },
  "status" : "OK"
}

```

([Lien](#))

Les données renvoyées par Google Places contiennent beaucoup plus d'informations que nécessaire mais pourront être utiles si nous souhaitons ajouter plus de fonctionnalités dans un temps futur. Combinée avec OpenStreetMap, les 2 APIs permettant du calcul géolocalisé, nous pourrions efficacement localiser les restaurants les plus proches de nos pistes cyclables.

# Technologies utilisées

Pour la réalisation de ce projet, nous avons décidé d'utiliser des technologies modernes et adaptées à nos besoins. Notre stack technique se compose donc de 3 éléments principaux :

- **TypeScript** comme langage de programmation
- **Postgresql** et **Neo4J** comme base de données

Nous allons donc décrire un peu plus en détail nos choix ainsi que les raisons qui nous ont poussés à les choisir.

## Langage de programmation

Pour ce qui concerne les langages de programmation, nous partirons sur le **Typescript**. Cette décision s'oriente autours de 6 points suivants :

1. Un écosystème fort : Typescript étant lui même un superset de NodeJS, il bénéficie d'une communauté extrêmement large
2. Son typage fort permet d'éviter la plupart des problèmes que nous pouvons rencontrer en Javascript pur
3. Le langage évolue rapidement, ce qui nous permet d'accéder aux dernières nouveautés dont l'optimisation
4. L'équipe a déjà travaillé en Typescript et maîtrise le langage, réduisant donc la dette technique
5. Sa syntaxe simple permet à un nouveau collaborateur d'intégrer rapidement le projet
6. Possède des intégrations avec les outils que nous utiliserons, dont les connecteurs de base de données

## Base de données

### PostgreSQL

Postgresql est une base de données SQL moderne permettant de stocker des données dans des tables relationnelles permettant de créer des liens entre ces différentes données. Postgresql est donc notre meilleur candidat afin de stocker les données des restaurants qui seront structurées de manière identique. Des alternatives comme MySQL ou MariaDB n'ont pas été retenues parce que trop

anciennes. L'approche moderne de Postgres lui permet d'être facilement mis à l'échelle grâce à ses stratégies de répliquions et de sharding simples. Enfin, parce que Postgresql est le logiciel de gestion de base de données SQL le plus utilisé aujourd'hui, il possède une grande communauté et donc un grand support, peu importe les plateformes.

## **Neo4J**

Neo4J est une base de données structurant les données sous formes de graphes, avec des noeuds et des relations. Elle permet de stocker des données aux nombreuses relations, ce qui sera parfait pour stocker nos itinéraires ainsi que les restaurants environnants. Neo4J permet d'effectuer des opérations géospatiales ce qui nous aidera énormément lors de la création d'itinéraires. Lui aussi leader dans son domaine, sa documentation complète et les nombreux exemples font de lui le candidat idéal pour notre utilisation. Depuis sa version 4.0, Neo4J permet à ses utilisateurs d'utiliser une politique de sharding afin de permettre une plus grande stabilité et une plus grande extensibilité.