

Technologies NoSQL

Projet de fin de module

J'ai choisi MongoDB car il semble le système de gestion de base de données NoSQL le plus populaire, et par conséquent le plus documenté. De plus, il est réputé pour pouvoir faire des recherches avancées de type géospatiale. La gestion de données GeoJSON, l'existence de l'index 2dsphere, et d'opérateurs comme \$near, sont une qualité apparente de MongoDB pour ce sujet, mais je n'ai malheureusement pas réussi à les exploiter et me suis rabattue sur une solution bien moins riche.

I. Installation de MongoDB

Lancer le script « installation.sh ». Puis, lancer « mongo » dans le terminal et taper « use nyc » pour créer et utiliser, à partir de maintenant, une base appelée nyc.

II. Télécharger roboMongo

Se rendre à l'URL suivante : <https://robomongo.org/>.

Vérifier qu'il s'agit de la version Linux.

Une fois le .tar téléchargé, l'extraire et se rendre dans Téléchargements/robomongo-0.9.0-linux-x86_64-0786489/bin/ et lancer l'exécutable robomongo.

Se connecter au port par défaut à partir de l'icône des ordinateurs en haut à gauche.

Ouvrir un nouveau shell à partir d'un clic droit sur « New Connection » dans l'arborescence.

III. Ma description d'un bon endroit où m'installer

Ma priorité est de m'installer à proximité d'une station de métro, car c'est mon mode de transport en commun favoris. Autrement, je me déplace à pied. Sinon, étant donné que je n'ai pas encore trouvé de travail, mes exigences en termes de loisirs sont d'être le plus proche possible du meilleur studio de pole dance de la ville « Body & Pole » pour faire du sport et du meilleur parc pour sortir mon chien « Washington Square Park Small Dog Run ».

IV. Téléchargement des jeux de données

Je télécharge donc les données de géolocalisation des stations de métro et des parcs et espaces verts à partir du site NYC OpenData. Lancer le script « telechargement.sh ».

V. Import des données dans MongoDB

Lancer le script « import.sh ». En faisant un « refresh » à partir d'un clic droit sur l'arborescence dans robomongo on devrait voir les tables apparaître.

VI. Requêtes mongoDB

//Précision de la base utilisée

use nyc

//Creation d'une sous-table à partir de la table des stations de métro, en isolant leurs coordonnées

```
db.subways.aggregate([
  [{
    $project: {
      NAME:1,
      x:{
        $substrBytes: [ "$the_geom",7,18 ] },
      y:{
        $substrBytes: [ "$the_geom",26,17 ] }
    },
  },
  {
    $out: "subwaysStations"
  }
])
)
```

//Conversion des coordonnées de String en Float

```
db.subwaysStations.find( { 'x' : { $type : 2 } } ).forEach(
function (e) {
  e.x = parseFloat(e.x);
  db.subwaysStations.save(e);
});
db.subwaysStations.find( { 'y' : { $type : 2 } } ).forEach(
function (e) {
  e.y = parseFloat(e.y);
  db.subwaysStations.save(e);
});
```

//Creation d'une sous-table à partir de la table des parcs, en isolant leurs coordonnées

```
db.parks.aggregate([
  [{
    $project: {
      PARK_NAME:1,
      x:{
        $substrBytes: [ "$the_geom",16,18 ] },
      y:{
        $substrBytes: [ "$the_geom",34,17 ] }
    },
  },
  {
    $out: "parksEntries"
  }
])
)
```

```

//Conversion des coordonnées de String en Float
db.parksEntries.find( { 'x' : { $type : 2 } } ).forEach(
function (e) {
    e.x = parseFloat(e.x);
    db.parksEntries.save(e);
});
db.parksEntries.find( { 'y' : { $type : 2 } } ).forEach(
function (e) {
    e.y = parseFloat(e.y);
    db.parksEntries.save(e);
});

//Calcul de la distance des station de métro au studio
« Body&Pole » à partir de ses coordonnées (trouvées à l'extérieur
de NYC OpenData car ce genre de studio n'y est pas renseigné)
db.subwaysStations.find().forEach(
function (e) {
    e.distPoleStudio = Math.sqrt((e.x-(-73.991701))*(e.x-(-
73.991701))+(e.y-(40.745972))*(e.y-(40.745972)));
    db.subwaysStations.save(e);
});

//Tri des stations de métro de la plus proche à la plus éloignée
du studio et sélection des 20 stations les plus proches du studio
db.subwaysStations.find().sort( { distPoleStudio: 1 } )
db.subwaysStations.aggregate(
    { $limit : 20 },
    { $out : "topSubways" }
);

//Recherche des coordonnées du meilleur parc à chien
db.parksEntries.find(
{ PARK_NAME: { $regex: /. *Washington Square.*/, $options: "i" } }
)

//Calcul de la distance entre le parc à chien et le top 20 des
stations de métro
db.topSubways.find().forEach(
function (e) {
    e.distPark = Math.sqrt((e.x-(-73.99658662665607))*(e.x-(-
73.99658662665607))+(e.y-(40.72965140215158))*(e.y-
(40.72965140215158)));
    db.topSubways.save(e);
});

//Tri du top 20 des stations de métro, cette fois-ci en fonction
de la distance au parc
db.topSubways.find().sort( { distPark: 1 } )

//La station gagnante est Astor Pl, je souhaite trouver au plus
près de cette station qui se trouve d'ailleurs à 5min à pied du
parc et 11min du studio en transport en commun.

```