

Rapport du Projet :

Mongo App – A NoSQL project

|  |  |
| --- | --- |
| Cours | Advanced Topics on NoSQL databases  Problem to solve |
| Groupe | 4 |
| Majeure | IBO-2 |
| Membres de l’équipe | TAZI Maxime  RAMOS Miguel  GOUJET Guillaume |

# TABLE DESMATIÈRES

Introduction 2

Resume du projet and queries 2

Liste des outils utilisees 2

ELABORATION du Projet 3

nosql database : mongo db 3

implementation 4

queries 5

conclusion 10

annexes facultatives #

# Introduction

###### résume du projet and queries

Notre projet consistait à créer une application, sur le langage de notre choix, sur une base de données NoSQL étudié pendant les cours.

Nous avons donc créé une application web codé en full JS. Cette application sert d’interface graphique pour la base de données NoSQL Mongo DB.

QUERIES :

* Rechercher des villes basées sur des paramètres (name, country, population, timeZone)
* Rechercher les villes les plus proches de chez vous basé sur vos coordonnées (longitude, latitude)
* Ajouter une ville à la base de données
* Filtrer vos recherches avec le nom de la ville
* Trier les villes par population
* Affichage très simple des villes sous la forme d’un tableau

###### LISTE DES OUTILS

* Front-end : ReactJS
* Back-end : NodeJS
* Design : Bootstrap, CSS
* Base de données : Mongo DB, GUI : Studio 3T

# elaboration du projet

##### NOSQL DATABASE : MONGO DB

On a choisi Mongo DB parce qu’on avait déjà une certaine expérience avec cette base de données. Le model de Mongo DB est plus orienté objet que Cassandra DB. On aurait pu choisir ElasticSearch mais son utilité pour l’analyse de données n’aurait pas été utilisé pour le projet qui se focalise plus sur les requêtes.

##### Implementation

Au niveau du serveur :

Pour créer un serveur web, où le site pourrait s’héberger, nous avons utilisé Node JS avec le module Express pour gérer les routes (endpoints).

En voici quelques exemples :



Figure 1: Search all cities endpoint (name, country … parameters)



Figure 2: Search near cities endpoint (lng and lat parameters)

Donc chaque rêquete était gérée par un endpoint avec des paramètres.

Le principe était que l’on puisse récupérer les paramètres par l’utilisateur et de traiter ces paramètres avec le module mongoose qui permet de manipuler Mongo DB à partir du code JS.



Figure 3: Requete near géré par mongoose

Ainsi on pouvait accéder aux résultats des requêtes via une API.

Au niveau client :

Pour afficher les résultats des requêtes et les formulaires pour que l’utilisateur puisse renseigner les paramètres, nous avons utilisé le framework ReactJS.

ReactJS est basé sur des états de données qui changent en fonction d’évenements. Nous avons donc créé par exemple un état de données des villes, et pour chaque requête effectué (bouton submit appuyé) cela déclenche un événement qui appelle une fonction.

La fonction fait une requête http sur les endpoints que nous avons créés.

Voici par exemple la fonction qui gère l’événement où l’utilisateur souhaite chercher une ville dans Mongo DB

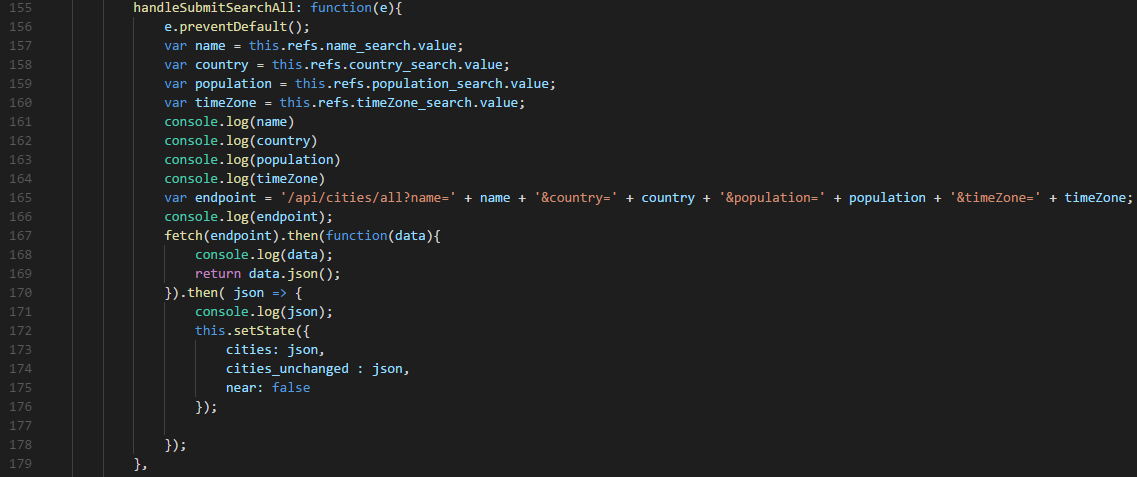


Figure 4: Event function fetching data from the api/cities/all endpoint

Une fois les données collectées à partir de l’API, on les affiche sous la forme d’un tableau en HTML :



Figure 5: HTML code to display data in the form of an array

Enfin, pour le design, nous avons utilisé la librairie Bootstrap et manipulé aussi du CSS.

Voici le rendu final de l’application :

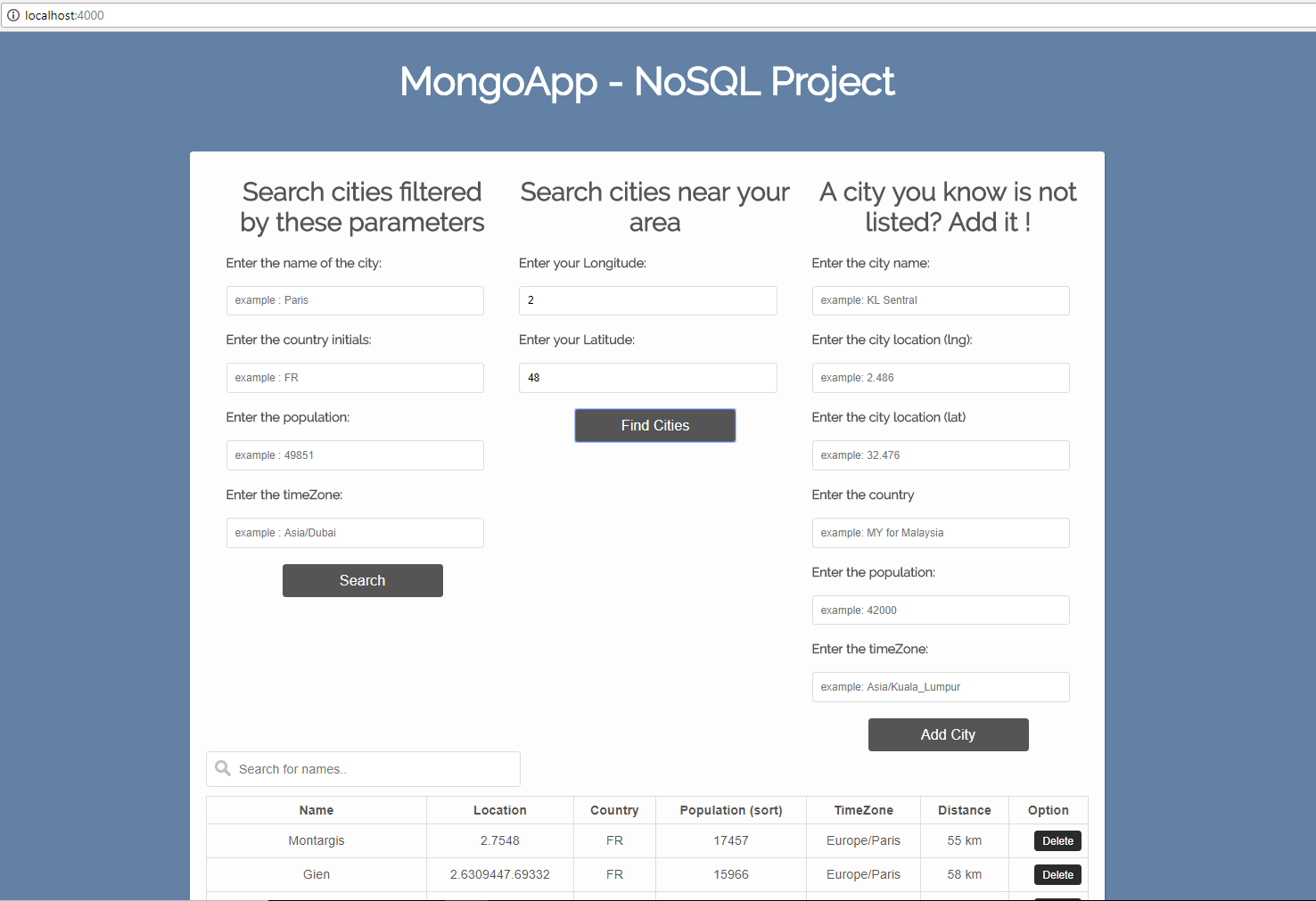


Figure 6: Final result of the web application

# ****Conclusion****

Ce projet nous as permis de nous intéresser aus sytèmes de base de données NoSQL, leurs avantages comme leurs inconvénients. Nous avons pu appliquer les technologies NoSQL dans un cas d’utilisation simple comme une application web orienté expérience utilisateur. Ce projet nous a aussi permis d’intégrer les connaissances acquises pendant les cours de Advanced Topics on NoSQL databases comme la manipulation des requêtes simples (filters) ou aggregates.

La suite de ce projet pourrait être de réussir à ce que l’application web propose plus d’interactivité (requêtes) avec l’utilisateur, et pourquoi pas changer de base de donnés NoSQL comme ElasticSearch qui nous permettrait en plus d’analyser les données plus simplement et aussi de les visualizers avec des graphes.

GIT HUB OF THE PROJECT : https://github.com/MiguelRamosF/AppMongo

