

	Cycle ingénieur 2^{ème} année NoSQL TP2 : Requêtes mongodb	
	<i>EISTI</i>	
	<i>Matière : BDD Avancée</i>	<i>Échéance : 9 octobre 2018</i>
	<i>Nombre de pages : 2</i>	

1 Requêtes

Exercice 1.

- a. Trouver le nombre de crimes impliquant un couteau ou une arme à feu.
- b. Sélectionner le nombre de cambriolages de juin à août 2018.
- c. Afficher les 3 crimes les plus anciens.
- d. Renommez "Cluster 6" en "Cluster 666".
- e. Afficher le nombre de crimes impliquant un vol.
- f. Afficher, par district, le nombre de crimes.
- g. Afficher, par moment de la journée, le nombre de crimes à arme à feu.
- h. Afficher, par moment de la journée, le nombre de harcèlements (SEX ABUSE).
- i. Afficher, par type de crime (offense)), le nombre de crimes.
- j. Afficher le nombre de crimes, pour les vols à la tire (ROBBERY) et les incendies volontaires (ARSON).
- k. Afficher la moyenne de crimes, par district.
- l. Afficher le pourcentage des incendies volontaires en août.
- m. Trouver le district à la plus forte criminalité.
- n. Supprimer les crimes relatifs aux incendies volontaires.
- o. Compter le nombre de crimes commis dans un rayon de 500m autour de la maison blanche (-77.03652979999998, 38.8976763). Le rayon de la terre est de 6371 km.
- p. Quels sont les types de ces crimes ?
- q. Afficher les 10 crimes commis les plus proches du Capitole (-77.009003,38.889931).

2 Interactions avec une application

Nous allons ici utiliser restHeart, une API permettant d'attaquer une base mongodb par service REST (cf cours J2E). Vous téléchargerez le jar ici. Pour simplifier l'exercice, nous utiliserons le jar avec les paramètres par défaut (pas d'authentification, port HTTP 8080, etc.). Pour une configuration plus poussée, c'est par là.

L'objectif ici est d'attaquer la base de données le plus simplement possible, sans passer par l'installation d'un serveur web d'un certain langage et du driver mongodb correspondant. Nous ne réaliserons que des requêtes d'interrogation de données (GET).

Exercice 2.

- a. Pour démarrer le service :
`java -Dfile.encoding=UTF-8 -server -jar restheart.jar`
Attention, le port par défaut du service est 8080, prenez garde à votre tomcat ou tout autre logiciel susceptible d'utiliser ce port.
- b. Vérifions le bon fonctionnement : dans un navigateur web, saisir les urls suivantes, À quoi correspondent les données affichées ?
`http://localhost:8080`
`http://localhost:8080/commissariat`
`http://localhost:8080/commissariat/crimes`
- c. Pour cette dernière requête, que remarquez-vous ?
- d. Ajouter en fin d'url `?count=true`, que remarquez-vous ?
- e. En utilisant les propriétés de pagination, afficher les 50 premiers crimes.
- f. En utilisant les propriétés de pagination, afficher les 50 premiers crimes, triés par district décroissant.
- g. Afficher les 50 suivants.
- h. Afficher 80 crimes du district 6.
- i. Compléter la requête précédente pour n'afficher que les types de crime.
- j. En utilisant l'API fetch de javascript, écrire une page html et un fichier javascript qui va exécuter la requête précédente et n'afficher qu'une seule fois les types de crimes.