

# TP2 - MapReduce et Index

---

La semaine dernière, nous avons utilisé une structure un peu trop complexe et ne reflétant pas du tout ce que vous pourrez trouver plus tard. Voici une structure de données plus réelle que nous allons utiliser pour ce TP. Cette structure représente une collection de statistiques où un document représente une action utilisateur :

```
{
  username : String,
  type : String,
  date : Date,
  device : String,
  website : String
  time : Number,
  category : String,
  article : String
}
```

Voici la liste des différents types possibles :

- **TIME** : Durée de la navigation. Il peut en avoir plusieurs par utilisateurs s'il s'est rendu plusieurs fois sur l'un des sites/articles. La durée en ms est stockée dans **time** .
- **READ** : Indique que l'utilisateur lis un article. L'article est indiqué par le champ **article** et, s'il vient d'une catégorie, c'est indiqué par le champ **category** .
- **CLICK\_CAT** : Indique que l'utilisateur a cliqué sur une catégorie pour, par exemple, voir les articles qu'elle contient. La catégorie est indiquée par le champ **category**

Vous trouverez, dans le dossier de partage, un fichier `initScriptTP2.js` que vous devrez exécuter dans mongobooster après avoir créé une nouvelle base de donnée.

## MapReduce

---

Vous devrez réaliser les requêtes mapReduce suivantes :

- Quel est le temps moyen de navigation par utilisateur. Triez le résultat du plus grand au plus petit.
- Quel est le temps moyen de navigation par site. Triez le résultat par ordre alphabétique.
- Quels sont les articles les plus consultés pour les `iphones` ?
- Quels sont les sites les plus visités ? (les stats de type `TIME` semblent être les plus judicieuses)
- Quels sont les sites avec le meilleur temps de navigation pour les 10 premiers enregistrements de la collection `stats` ? (utilisation du `limit`)

## Index

---

- À partir des requêtes précédentes et d'après vous, quel sont les index à créer pour optimiser nos requêtes précédentes ? Vous pouvez effectuer un simple `find()` sur la collection en fonction des queries précédentes couplé à un `explain()` pour vous aider.
- Créez un index de type texte sur les champs `article` et `category` puis effectuez la requête suivante :
  - Combien il y a de lignes de stats pour le type `READ` contenant les mots `politique` `tempore` ?

# Bonus

---

S'il vous reste du temps en fin de TP, vous pouvez terminer le TP 1. La correction de ce dernier sera mise en ligne demain.