



# Client Serveur 2021

---

## Explications (à lire attentivement avant de se lancer !)

---

Vous disposez de deux heures pour répondre aux questions suivantes. Pour chaque réponse que vous donnez, les points seront bien entendu sur l'argumentation. Oui ou Non ne rapporte pas de points.

Ne vous laissez pas influencer par **Filibert**, il se peut qu'il ait tort parfois, ou alors peut-être qu'il a toujours raison.... Seul l'avenir le dira. A chaque question impliquant **Filibert**, vous allez devoir argumenter à charge ou à décharge en fonction de vos arguments.

Quoiqu'il en soit, il y a souvent plusieurs "bonnes réponses", laissez une réponse vide parce que vous n'êtes pas sûrs n'est pour cet examen pas forcément une bonne idée.

Ce qui comptera le plus c'est la justesse de votre raisonnement même si moi je suis plus ou moins d'accord avec les propos de Filibert en fonction des questions. Si votre raisonnement et vos justifications sont justes vous aurez les points.

## Q1 - Modification simple modèle de données NoSQL

---

Une multinationale se rapproche de vous et est très intéressée par votre application.

Cependant, elle aimerait collecter des données supplémentaires que vous n'avez pas déjà incluses dans votre modèle.

Elle vous demande donc de voir comment vous pourriez répondre à leur besoin en adaptant votre modèle.

Voici les informations qu'ils souhaitent:

- L'adresse d'un utilisateur (ville et nom de rue suffiront)
- Le sexe de l'utilisateur (pour pouvoir faire des statistiques et voir que les femmes s'acquittent de leurs dettes plus rapidement que les hommes!)

Expliquez dans quelle(s) collection(s) vous allez rajouter ce(s) champ(s) ET expliquez comment cela va vous permettre de répondre au besoin énoncé.

## Q2 - Modification complexe modèle de données NoSQL

---

Ils aimeraient également ajouter à votre application des composantes sociales. Pour cela, ils aimeraient que les utilisateurs puissent avoir des listes de contacts nommées (par exemple une liste nommée 'famille' avec leur maman et leur sœur et une liste nommée 'coloc' avec les larrons avec qui il partage son appartement) .

Leur cahier des charges est le suivant :

- Une liste de contact possède un nom et une liste d'utilisateur de la plateforme Okapi.
- Un utilisateur peut avoir plusieurs listes de contacts.
- Il n'y a pas de limites aux nombres de contacts qu'un utilisateur peut avoir dans une liste, mais on peut facilement imaginer qu'ils n'en auront qu'une poignée chacune.

Expliquez les modifications/ajouts que vous allez faire à votre model de données NoSQL actuel pour répondre à ce besoin.

## Q3 – Explications Filibert

---

**Filibert** est le petit nouveau qui vient d'arriver dans votre équipe, il n'a jamais fait de NoSQL. Il s'approche donc de vous tout timide et vous demande de lui expliquer la chose suivante :

Il se demande pourquoi vous avez stocké toutes les informations des dettes dans la collection Bill au lieu de simplement ajouter une collection "Requests" et faire une jointure entre ces deux collections (Bill et Requests) dans une troisième collection.

Expliquez lui votre choix.

## Q4 – Challenge Filibert

---

*Maintenant que vous avez tout bien expliqué à Filibert, il a cogité toute la nuit.*

*Filibert s'interroge sur le sens de la vie, mais aussi sur d'autres choses banales comme pourquoi Iron Man est mort dans le dernier Avengers... Bref beaucoup de questions inutiles ici !*

Mais il a aussi une vraie question pour vous :

Et si on modifiait un peu notre base de données pour inclure directement les utilisateurs dans chacune de nos occurrences de documents Bill (oui oui avec tous ses champs, enfin Filibert il ferait bien ça lui) et on en profiterait pour supprimer la collection "User" ? Ça pourrait être bien non ? Ça nous ferait une collection en moins à gérer.

Que répondez-vous à Filibert ? (pas 42, ni de réponse sur la perte tragique de Tony Stark)

## Q5 – Design Pattern

---

*Filibert est très branché Design Pattern, il a un grand poster au-dessus de son lit où l'on peut lire : "Un patron de conception est la meilleure solution connue à un problème de conception récurrent"*

Filibert est également de nature très inquiète, il vous demande donc au détour d'un café comment vous comptez gérer la migration de vos données à la suite de votre prochaine mise en prod qui va nécessiter de rajouter des informations dans votre base de données.

Expliquez en détail à ce cher Filibert ce que vous prévoyez pour limiter au maximum les effets indésirables sur votre application.

## Q6 – Indexation

---

Filibert vient de terminer sa tâche de revue des indexs existants.

Il vous demande ce que vous pensez de remplacer les indexs suivants :

- Collection Bill : {requests.createdAt : 1}
- Collection Bill : { requests.debtor : 1}

Par l'index suivant :

- Collection Bill : {requests.debtor : 1, requests.createdAt : 1}

Qu'en pensez-vous ? Donnez votre avis à Filibert

## Q7 – Requêtage

---

Pour chaque requête, veuillez détailler pour chaque étape l'opérateur d'agrégation à utiliser (match, sort, group, ...) et indiquer avec une phrase française sur quel(s) champ(s) vous voulez l'appliquer et pour obtenir quel résultat. L'écriture de la requête exacte (format mongo) rapportera 1 point de plus par requête.

- ☐ Comment feriez-vous pour récupérer le total des factures d'un utilisateur donné (par son id)?
- ☐ Comment feriez-vous pour trouver la liste des débiteurs de la plus grosse facture (€€) encore non totalement remboursée?

## Q8.1 – Mise en perspective du cours

---

Si vous deviez refaire ce projet avec un choix libre sur les technos à utiliser, resteriez-vous sur une base NoSQL (MongoDB) ou passeriez-vous sur une base relationnelle ? Pourquoi ?

## Q8.2 – Mise en perspective du cours

---

Qu'avez-vous retenu de MongoDB si vous deviez la décrire en quelques mots ?

## Q8.3 – Mise en perspective du cours

---

Qu'avez-vous appris sur les optimisations SQL (base relationnelles) ?