Equipe_1_03

Rapport R4.09

SAE 4

Le présent document vous est présenté par l'**Équipe_1_03**, composé des membres suivants :

- XXXX
- LAURENS Thibaud
- YYYY
- ZZZZ

Celui-ci a pour but de vous expliquer la logique métier de notre serveur / application au travers de processus modélisés avec BPMN 2.

BPMN est un langage formel, standardisé, utilisé pour décrire des processus métiers. Cela permet aux entreprises de mieux visualiser leurs procédures internes afin de se comprendre elles-mêmes et de s'adapter aux changements, ainsi que de communiquer avec des tiers de manière standard.

Nous vous présenterons dans l'ordre les points définis dans le sommaire suivant :

Sommaire :

Contexte **2**
Diagrammes **3**
1) Affichage principal 3
2) Informations du livre 4
3) Recherche par texte 5
4) Recherche avancée 6
5) Recherche par mot-clé 7
6) Inscription & Connexion 8
7) Gérer ses bibliothèques personnelles 9
8) Gérer les livres dans les bibliothèques personnelles 10

Contexte

Dans le cadre de la SAE 4 “Réalisation d’une application complexe”, il nous est demandé de concevoir et développer une application reposant sur des micro-services communicants à destination de clients qui sont, eux aussi, des applications. Celle-ci se basera sur des API RESTful disponibles publiquement.

Notre application est basiquement une bibliothèque en ligne. Le concept est de mettre à disposition de nos clients (c’est-à-dire n’importe qui), en ligne et gratuitement, les informations importantes concernant une centaine de livres au minimum : titre, isbn, nombre de pages, date de parution, etc.

Ces données proviennent de plusieurs API publiques, à savoir *Open Library* et *Google Books* pour le moment. L’intérêt de mettre en commun les données de plusieurs API sur un même livre est de pouvoir combler les trous là où une API manqueraient de données. Par exemple, si Open Library ne connaît pas la date de parution d’un livre, il est probable que Google Books possède cette information pour compléter celles que nous donnons ultimement à nos clients.

Notre bibliothèque est utilisable sans connexion / inscription, dans quel cas les services proposés sont les suivants :

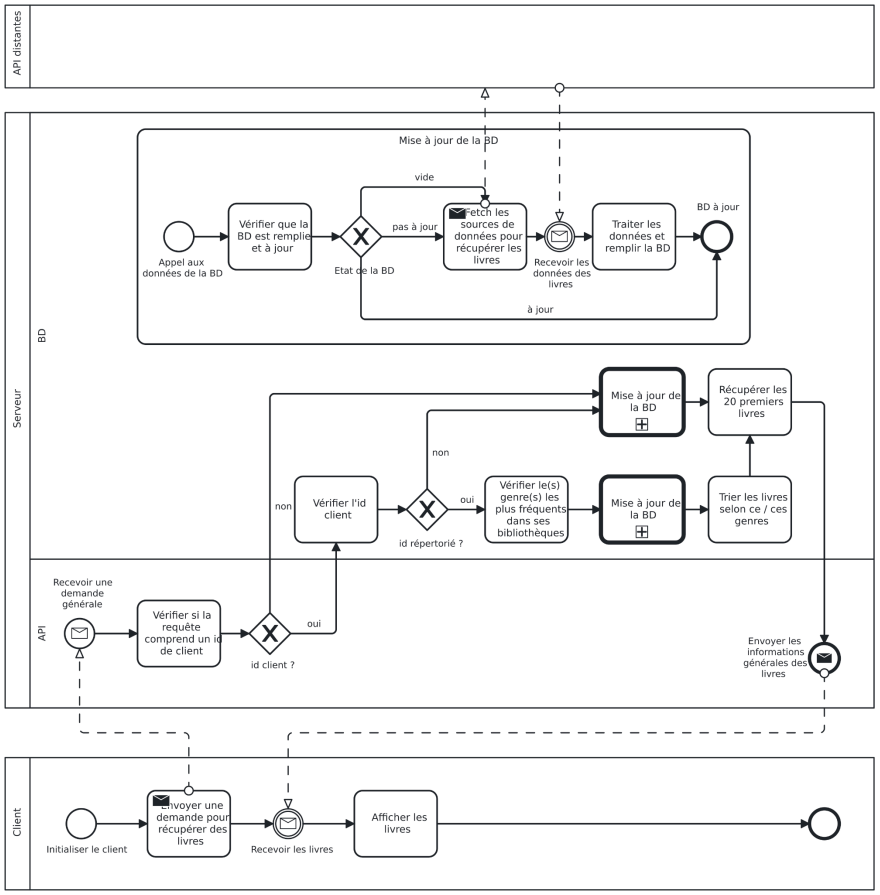
- Récupérer et unifier les données de 100 livres depuis les API publiques
- Donner les informations générales des 20 premiers livres recensés
- Donner les informations détaillées d’un livre spécifique, selon son ISBN
- Chercher des livres à partir d’une chaîne de caractères (vérifie le nom de l’œuvre et de l’auteur)
- Chercher des livres à partir d’un critère spécifique (auteur, genre, nom d'œuvre, année de publication, etc.)

Un système d’inscription / connexion facultatif est proposé afin de rajouter les fonctionnalités suivantes :

- S’inscrire et se connecter : les clients n’ont pas vraiment d’état “en ligne” ou “hors ligne” ; si les informations de connexion sont exactes on leur renvoie à la place leur identifiant interne, qu’ils pourront utiliser pour les points suivants
- Créer des bibliothèques personnelles, les supprimer et les renommer
- Ajouter un livre consulté dans l’une de leurs bibliothèques personnelles et l’en retirer
- Chercher des livres à partir d’une bibliothèque personnelle comme critère
- Améliorer la récupération d’informations générales des 20 premiers livres recensés en procédant selon les genres favoris des clients

Diagrammes

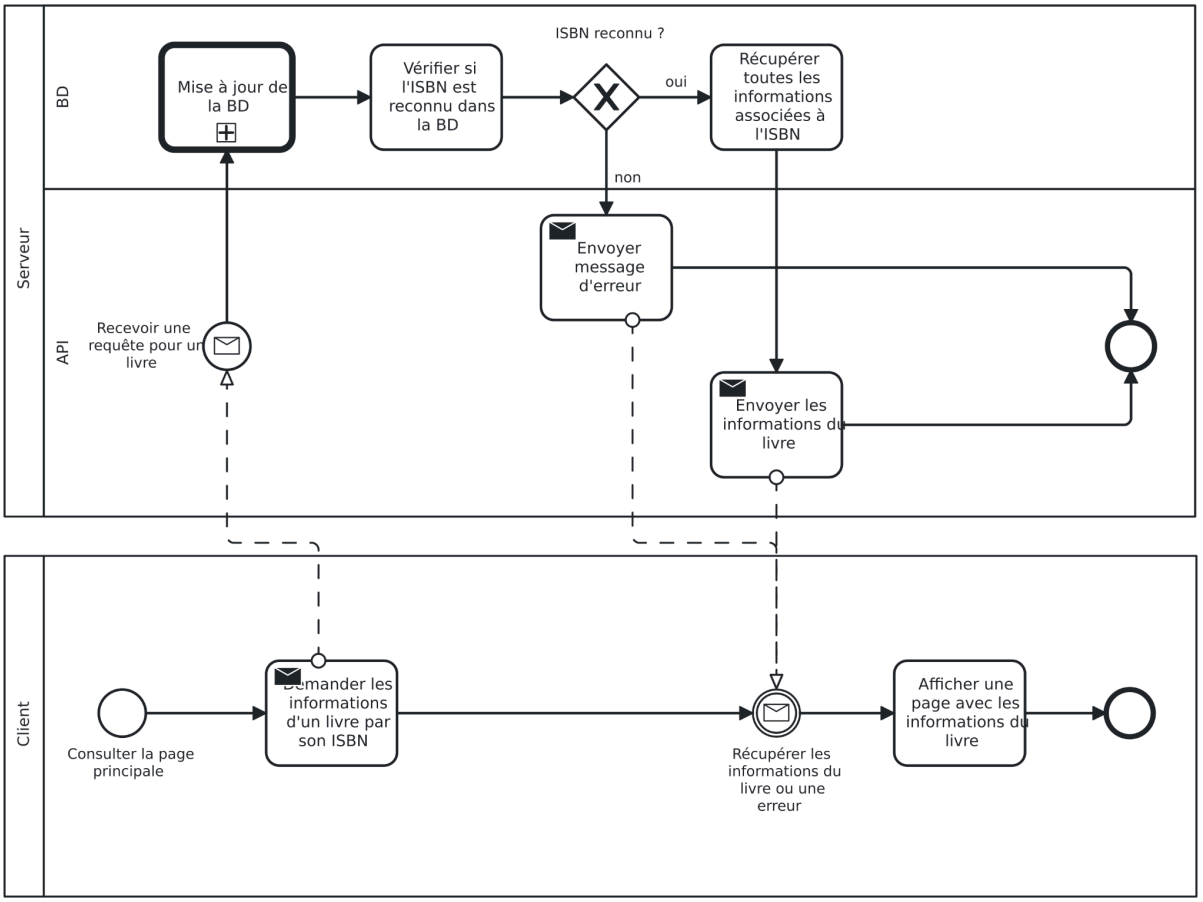
1) Affichage principal



La fonctionnalité de base de notre application. Le scénario nominal est de recevoir une demande du client pour les informations générales de livres sans id_client spécifié. Dans ce cas, comme pour toutes les autres requêtes impliquant de piocher dans les livres que nous recensons, nous commençons par mettre à jour la base de données si besoin. Ce sous-processus sera réutilisé dans de nombreux diagrammes. Cela fait, nous récupérons les informations demandées et les envoyons au client.

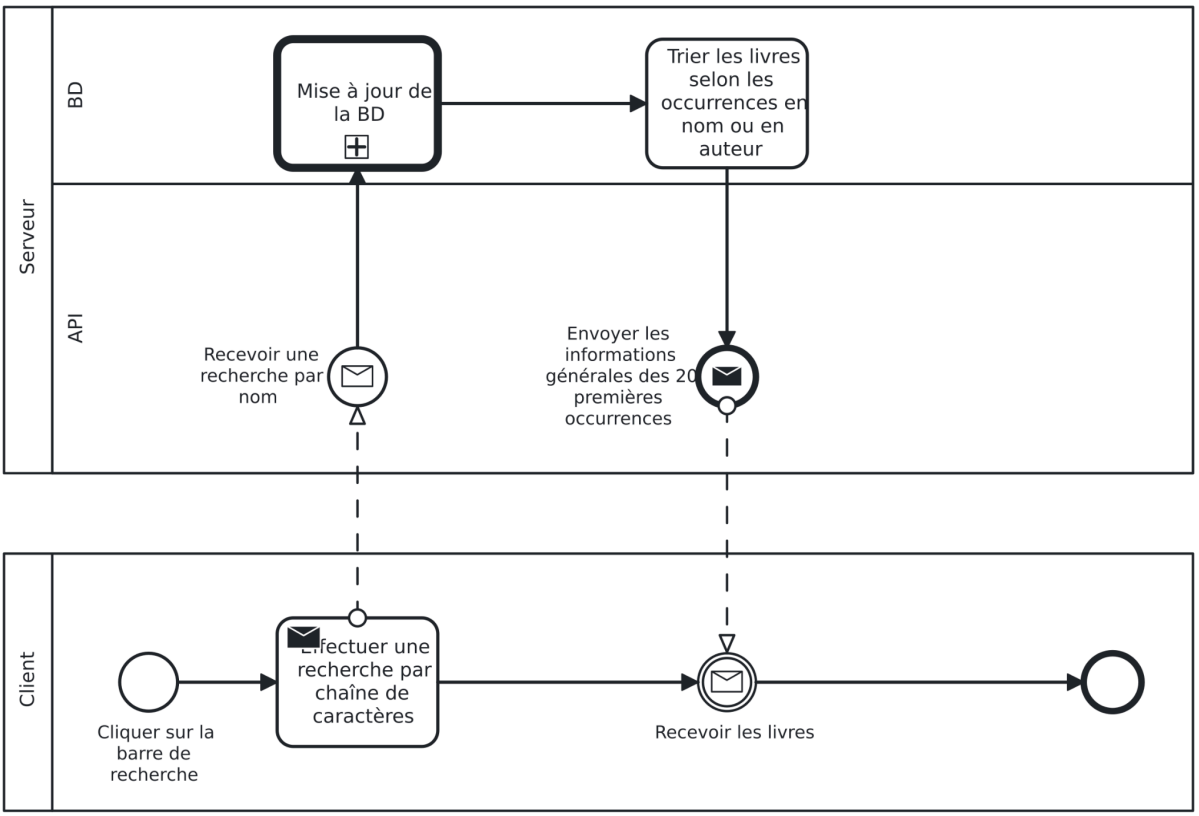
Si un id_client est spécifié et valide, on vérifie les bibliothèques du client afin d'en déduire ses genres favoris et de retourner des livres correspondant à ses goûts. Dans le cas où l'id_client n'est pas valide, nous procédons comme s'il n'avait pas été spécifié.

2) Informations du livre



Ce processus débute lorsqu'un client demande l'ensemble des informations d'un livre correspondant à l'ISBN passé en entrée. La requête étant centrée sur les livres que nous détenons en mémoire, on commence par mettre à jour si besoin la base de données. Cela fait, on vérifie la validité de l'ISBN donné. S'il l'est, on récupère tout ce qu'on connaît sur ce livre et envoie la réponse au client. Sinon, on lui retourne un message d'erreur.

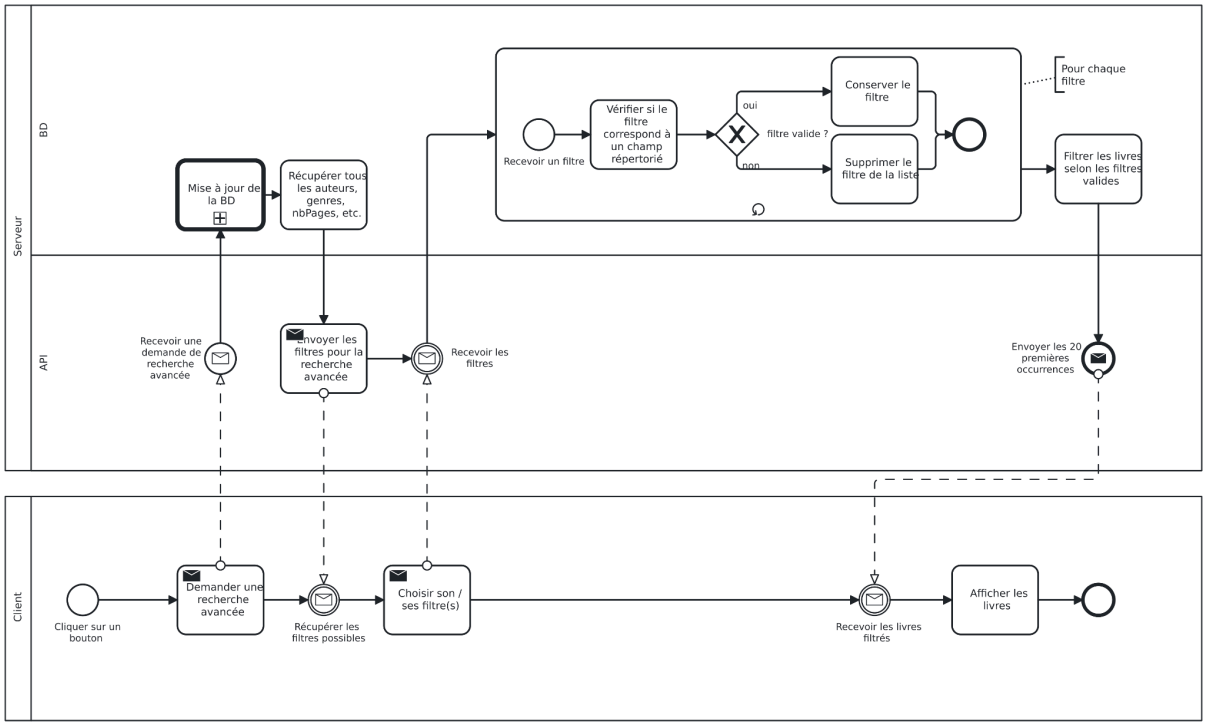
3) Recherche par texte



Le processus le plus simple de notre application, ne fournissant qu'un unique enchaînement d'actions et événements. Lorsque le client recherche des livres répondant à une certaine chaîne de caractères (d'une lettre à plusieurs mots), on commence par mettre à jour la BD si besoin, puis on vérifie pour chacun de nos livres si le nom de l'ouvrage ou celui de l'un de ses auteurs correspond à la chaîne de caractères. Ainsi, on filtre les livres pour ne renvoyer que ceux répondant à la requête, jusqu'à 20.

Si aucun livre ne correspond à la chaîne de caractères, le comportement du serveur reste le même : on renvoie simplement un texte au format JSON vide.

4) Recherche avancée

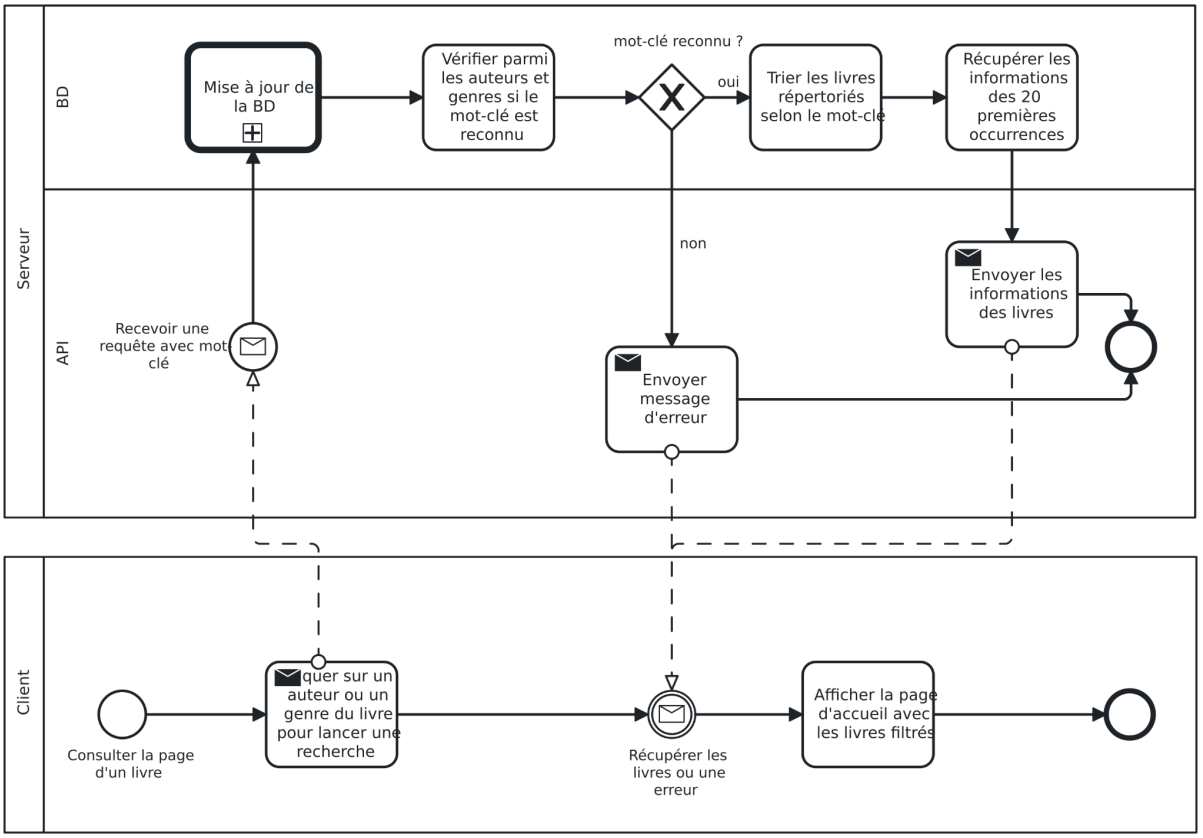


L'existence de ce processus métier constitue le gros de notre plus-value (hors fonctionnalités se basant sur l'authentification). Si un client souhaite réaliser une recherche avancée (basée sur des critères) sur les livres, la première étape est encore et toujours de mettre à jour notre BD si nécessaire, suite à quoi on fait l'inventaire de tous les auteurs existant dans la BD, tous les genres, les nombres minimum et maximum de pages, la décennie la plus ancienne et la plus récente, etc. et en fait part au client.

Ce dernier peut alors faire son choix parmi les filtres. Il peut en demander un seul, plusieurs, ou bien aucun.

Dans tous les cas, les filtres sélectionnés sont analysés un à un afin de s'assurer qu'il s'agisse bien d'un des filtres envoyés plus tôt. En effet, le client a beau connaître la liste des filtres autorisés, il reste libre de demander n'importe quoi ! Seuls les filtres valides sont conservés et sont utilisés pour trier nos livres. De la même façon que pour la recherche textuelle, on renvoie jusqu'à 20 correspondances, le cas où aucun livre ne correspond aux critères en revient juste à renvoyer un texte vide ; ce n'est pas un scénario exceptionnel ou une erreur. De même, si aucun filtre n'est soumis par le client, on retourne simplement les 20 premiers livres de la BD.

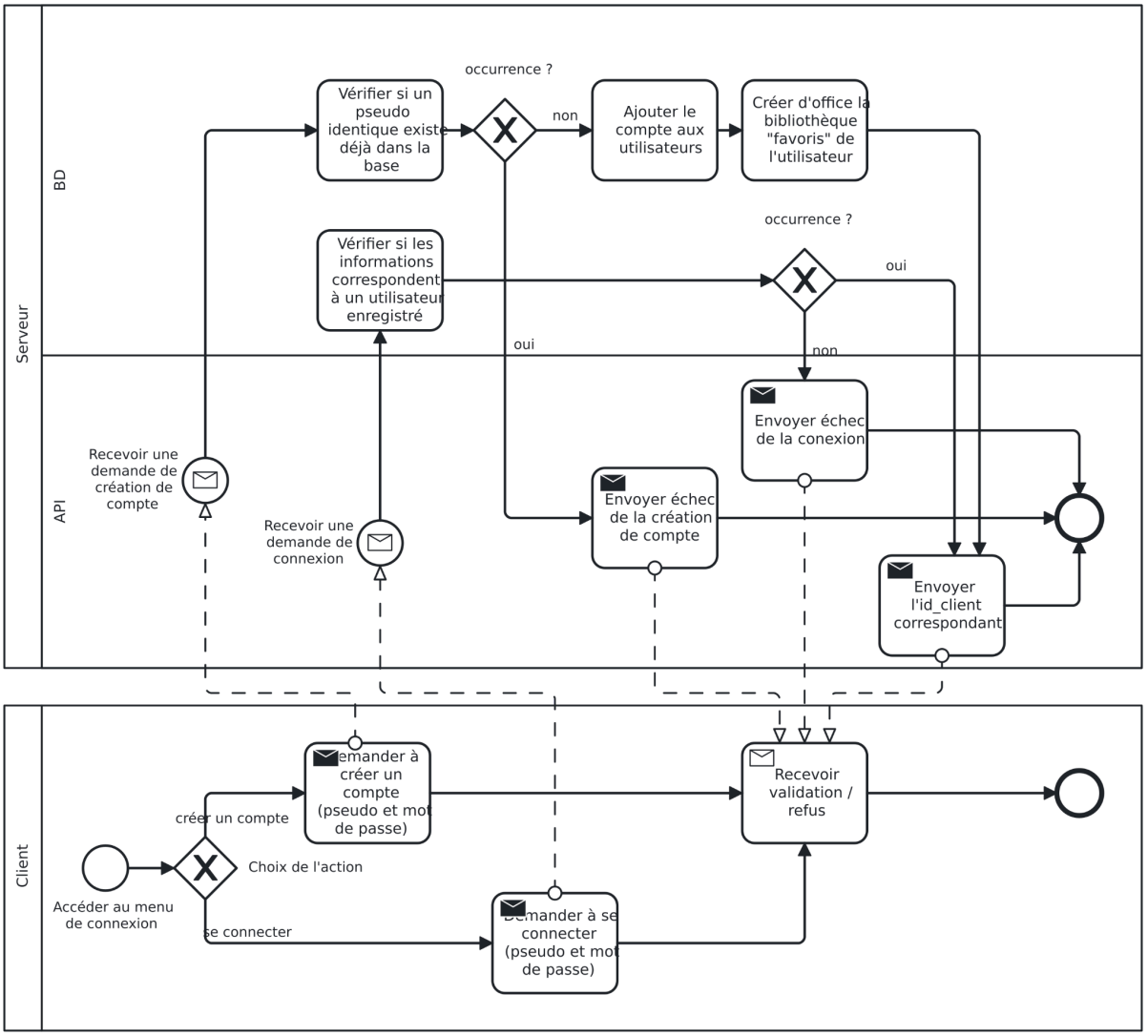
5) Recherche par mot-clé



Il s'agit sensiblement de la même chose que le processus métier précédent, à la différence près que ce processus-ci est plus simple et pensé pour un autre contexte. Ici, on attend de recevoir de la part de client une recherche de livres basée sur un et un seul mot-clé, non pas une liste de filtres. Ce mot-clé est pensé pour être utilisé en un clic sur le nom d'un auteur ou d'un genre. En définitive, ce processus est à mi-chemin entre "Recherche avancée" et "Informations du livre".

Comme toujours, on met à jour la BD si elle ne l'est pas, puis on s'assure que le mot-clé corresponde à un auteur ou un genre présent dans la base de données. Si oui, on retourne les informations générales de maximum 20 livres respectant ce mot-clé, sinon on renvoie un message d'erreur.

6) Inscription & Connexion

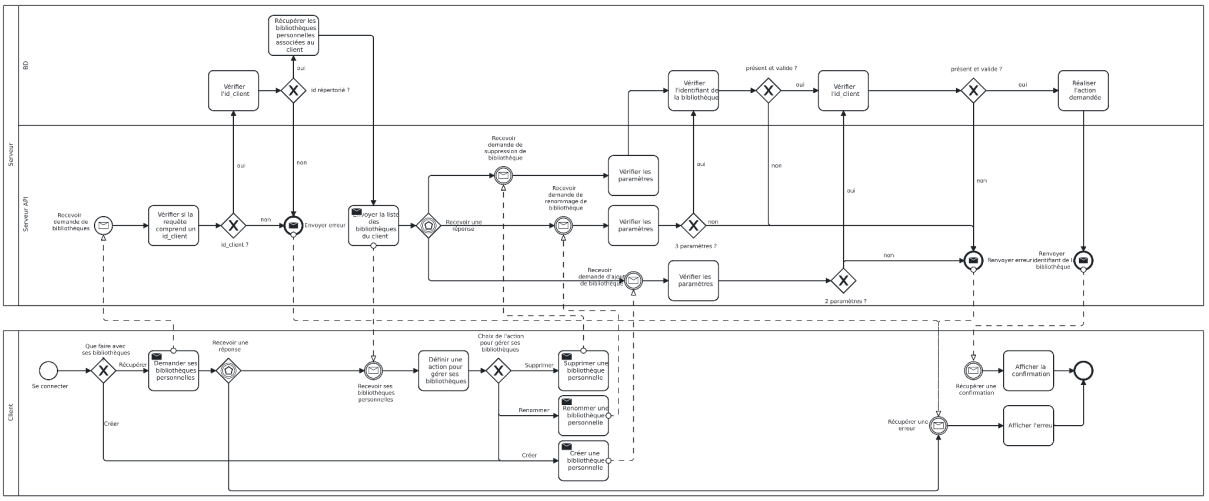


Ce processus est à la base de tous ceux qui viendront après. Comme il n'interagit pas avec les livres, nous n'avons pas besoin de vérifier si la BD doit être mise à jour ou non. Deux événements déclencheurs sont possibles du point de vue du serveur : le client demande soit à s'inscrire, soit à se connecter.

Dans le cas de l'inscription, on vérifie si le pseudo souhaité existe déjà dans notre BD. Si oui, une erreur est délivrée au client pour ne pas finir avec plusieurs clients au pseudonyme identique. Si non, on crée le compte en recopiant pseudo et mot de passe, puis on lui fabrique d'office une première bibliothèque personnelle : "favoris". On renvoie au client son identifiant pour l'informer du succès de l'opération.

En cas de connexion, on vérifie plutôt que le pseudo et le mot de passe donnés correspondent à l'enregistrement d'un client. Si cela est bien le cas, on envoie aussi son identifiant. Si les informations de connexion sont incorrectes, on retourne à la place une erreur.

7) Gérer ses bibliothèques personnelles



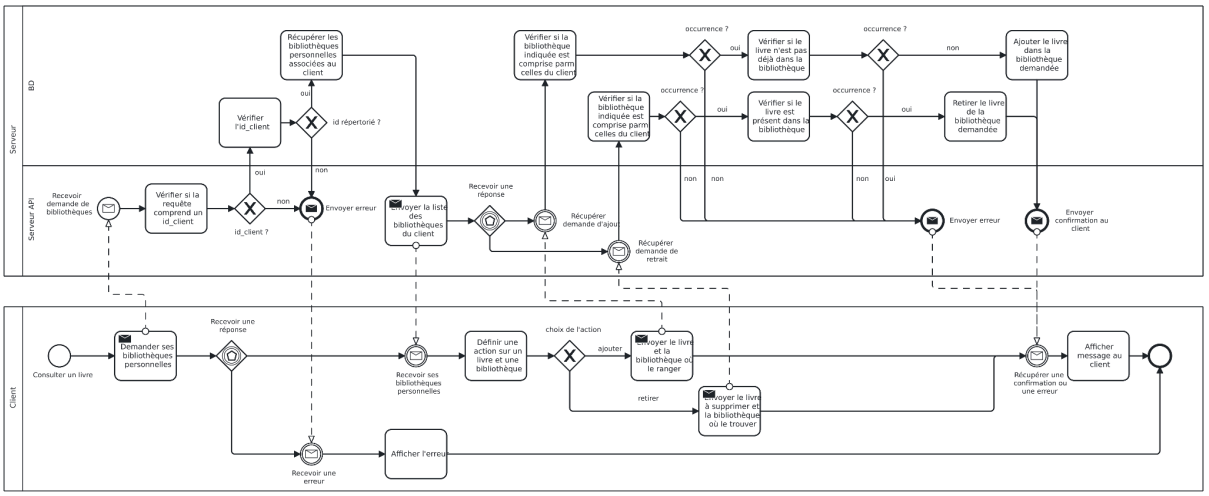
Le plus long diagramme de ce rapport couvre tous les aspects de la gestion d'une bibliothèque personnelle selon le système CRUD : créer, lire, modifier, supprimer.

Le client demande tout d'abord à connaître la liste de ses bibliothèques personnelles. Pour cela, il envoie au serveur son identifiant de client (en principe), ce qui est vérifié par le système. S'il est reconnu, on porte à sa connaissance l'ensemble de ses bibliothèques (identifiant et nom). Ces identifiants de bibliothèques lui sont nécessaires pour ensuite demander à en supprimer ou en renommer une. En revanche, cela est facultatif s'il s'agit de créer une nouvelle bibliothèque.

Nous permettons de cette façon à nos clients de créer, renommer et supprimer leurs bibliothèques personnelles. Pour faire ces requêtes, ils doivent dans tous les cas envoyer l'identifiant du client sur lequel agir. L'identifiant de la bibliothèque concernée est demandé pour renommer et supprimer (une bibliothèque qui n'est pas encore créée n'a pas d'identifiant), tandis que le nom de la bibliothèque est exigé pour créer et renommer une bibliothèque ; cette précision n'est pas demandée pour en supprimer une.Tous ces paramètres sont vérifiés au cours du processus.

Le client reçoit soit une confirmation comme quoi l'opération a réussi, soit un message d'erreur dont la nature dépend d'à quel moment l'erreur a été levée.

8) Gérer les livres dans les bibliothèques personnelles



Notre dernière logique métier réside en la possibilité d'ajouter un livre dans une bibliothèque personnelle d'un client et de le retirer. Quand un client en fait la demande, il indique en principe l'identifiant du client sur lequel agir (ce que retourne une inscription ou une connexion réussite). On prendra soin de s'assurer qu'il en envoie un, et aussi que l'identifiant corresponde à un de nos clients enregistrés. Si échec de l'une de ces deux conditions, on renvoie une erreur. Autrement, on récupère la liste des bibliothèques personnelles du client demandé et lui la fait parvenir.

On n'a plus qu'à attendre que le client choisisse un livre, une action à effectuer et une bibliothèque parmi la précédente liste. Pour autant, il peut toujours envoyer n'importe quoi, pour cela on vérifie que la bibliothèque personnelle demandée est comprise dans la liste précédemment envoyée. Pour ajouter un livre dans une bibliothèque personnelle, il ne doit pas déjà s'y trouver. Pour retirer le livre de la bibliothèque, c'est l'inverse. Si tout est bon, l'action est effectuée et une confirmation est transmise au client. Dans le cas où ce qu'il nous a fait parvenir n'est pas valide, une erreur est là aussi prévue.