BUT informatique - IUT - Nantes Université

2025-2026

[redacted], Thibaud Laurens, [redacted]

Rapport de

Modélisation par les automates des fonctionnalités d'une API de bibliothèque

SAE401 - Développement d'une application complexe

Module : Automates et Langages

1 - Introduction 2
2 - Présentation des différents microservices 2
A - Les microservices d'affichage 3
A.a - affichage de la page principale 3
A.b - affichage des données d'un livre 4
B - Les microservices de recherche 5
B.a - recherche simple 5
B.b - recherche avancée 6
C - Les microservices de sauvegarde de données 7
C.a - enregistrement/suppression d'un livre sur une bibliothèque personnelle 7
C.b - traitement des bibliothèques personnelles 9
3 - Automates et code 11
A - Les automates entre le code et l'analyse 11
B - Les limites de l'automate 11
4- Conclusion 12
5- Références **12**

1 - Introduction

Le projet que nous présentons est une API utilisable pour créer des applications de bibliothèques. L'objectif est de permettre à un client qui appelle l'API de collecter des livres issus de différentes API publiques en ligne et de les regrouper. Notre API récupère et stocke donc les informations liées à un certain nombre de livres dans une base de données et renvoie les informations demandées au client.
Dans ce rapport, nous allons présenter notre API à travers les différents microservices qu'elle propose, modélisés sous forme d'automates.
Le travail de création des automates a pour objectif de créer un modèle à suivre pour les développeurs. Ainsi, la création et la maintenance du code est facilitée grâce à une documentation complète et compréhensible.

2 - Présentation des différents microservices

Les différents microservices sont regroupés en trois catégories distinctes : l'affichage, la recherche et la sauvegarde de données.

Dans la suite du rapport, un certain nombre de ces microservices seront présentés. Les automates de tous nos microservices, présentés ou non, sont présents en annexe.

Pour comprendre les automates sur les pages suivantes, il est important de faire un point sur l'alphabet utilisé pour ces automates et la signification de certains caractères.
Chaque changement d'état est traduit par un nom logique.
Un ? avant un nom de changement d'état signifie que le client fait une requête.
Un ! avant un nom de changement d'état signifie que l'API répond au client.
Un Get signifie qu'il y a une requête pour des données .
Un Post signifie que l'API renvoie des données au client.
Un Close signifie que le client finit l'interaction avec l'API.

Les majuscules et espaces ne sont présents que pour la lisibilité et n'ont pas d'impact sur la logique.
La communication client-serveur passe par un seul canal, nous omettons donc volontairement d'expliciter le canal emprunté par les échanges.

A - Les microservices d’affichage

Les microservices d’affichage sont conçus pour rendre un certain nombre de données sans recherche explicite du client. Cela inclut l’affichage de la page principale et l’affichage des données d’un livre.

A.a - affichage de la page principale

Voici la modélisation de l’affichage de la page principale pour l’API :

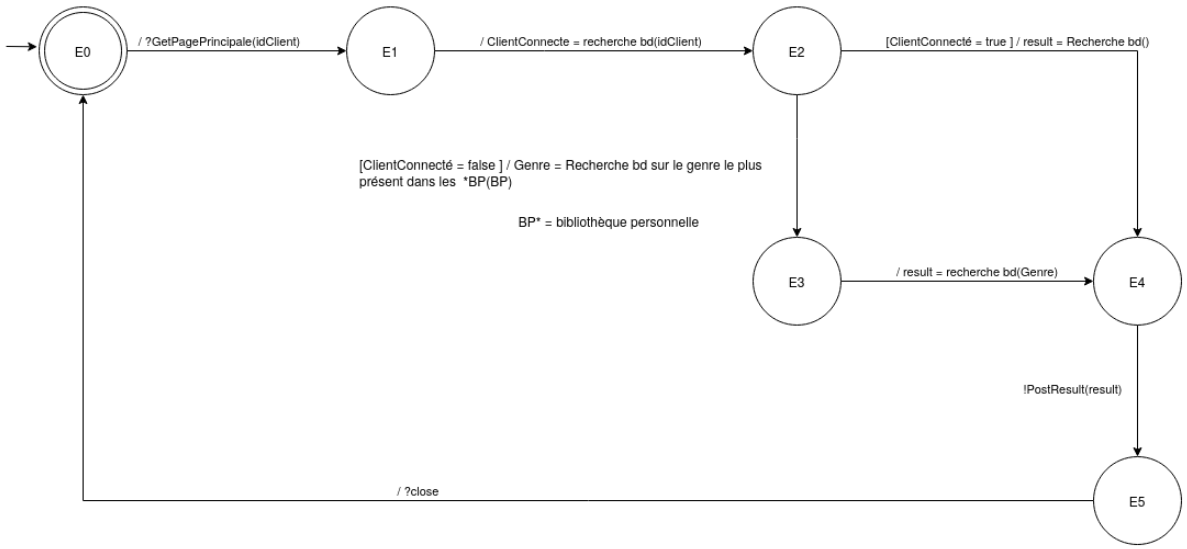

figure 1.

Fonction figure 1 :
. recherche bd() : récupère toutes les données en base de données liées au paramètre. Si il n’y a pas de paramètre, elle retourne les livres les mieux notés.
Recherche bd du genre principale des BP : rend le genre le plus présent dans les BP.

Spécification figure 1 :
Quand le client arrive sur le site :
- Si le client n'est pas connecté : l'API récupère dans la base de données les 20 livres les mieux notés et les envoie au client.
- Si le client est connecté : l'API va chercher les différentes bibliothèques personnelles du client dans la base de données afin de trouver le ou les genres les plus présents dans ses bibliothèques, puis elle récupère les 20 livres les mieux notés de ce(s) genre(s) dans la base de données.

A.b - affichage des données d’un livre

Voici la modélisation de l’affichage des données d’un livre pour l’API :

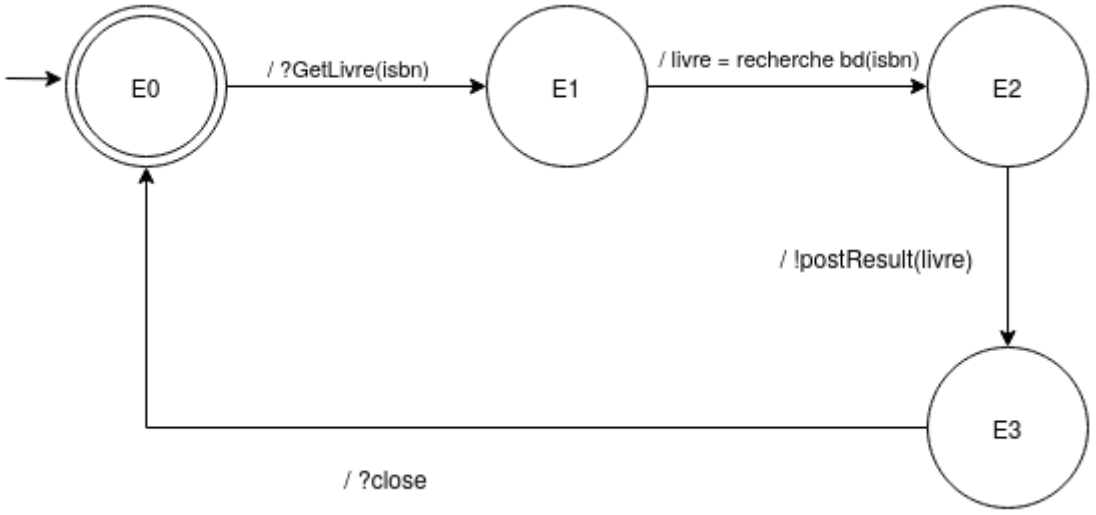

figure 2.

Fonction figure 2:
. recherche bd(isbn): rend toutes les informations de la bd sur le livre avec l’isbn correspondant

Spécification figure 2 :
Quand le client clique sur l'aperçu d'un livre, une requête est envoyée à l'API sur l'isbn de ce livre. Toutes les données du livre sont ensuite transmises au client via l'API pour l'afficher.

B - Les microservices de recherche

Les microservices de recherche sont conçus pour répondre à une recherche précise formulée par l'utilisateur. L'utilisateur peut donc faire une recherche simple sur le nom de l'auteur ou d'un livre. Il peut aussi faire une recherche avancée avec différents champs ou faire une recherche en cliquant sur un mot-clé issu d'une page.

B.a - recherche simple

Voici la modélisation de la recherche simple pour l'API :

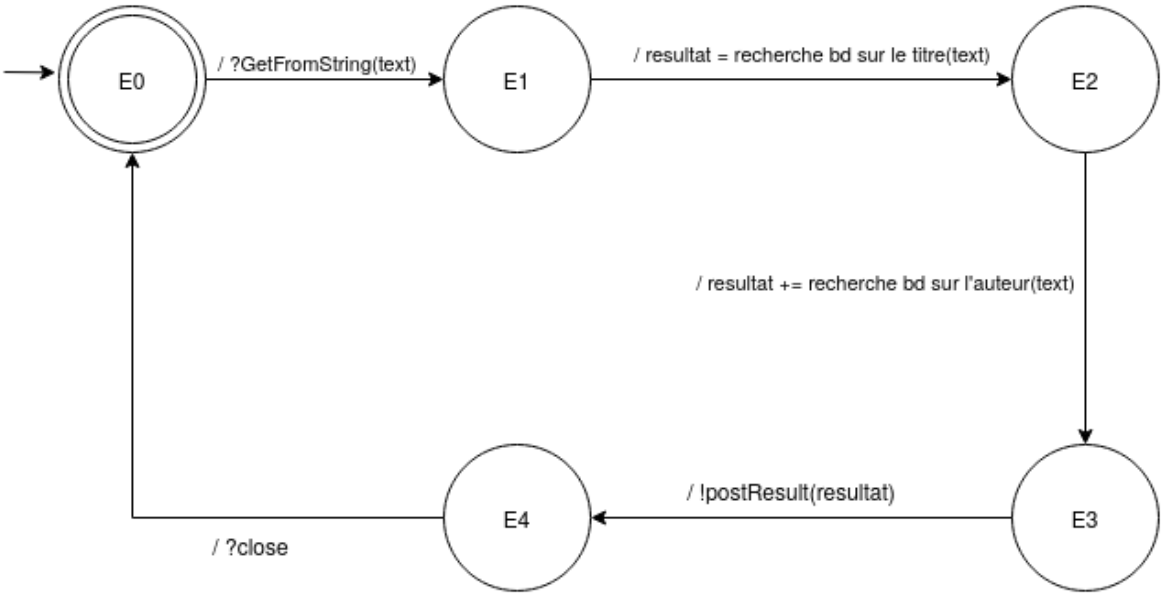

figure 3.

Fonction figure 3:
. recherche bd sur le titre(text) : rend tous les livre avec un titre correspondant à texte
. recherche bd sur l'auteur(text): rend tous les auteurs avec un nom correspondant à text

Spécification de la figure 3. :
Le client écrit dans la barre de recherche le nom d'un auteur ou d'un livre recherché. L'API reçoit le résultat de la recherche. Elle fait ensuite une requête à la base de données pour les livres/auteurs voulus et renvoie les résultats.

B.b - recherche avancée

Voici la modélisation de la recherche avancée pour l'API :

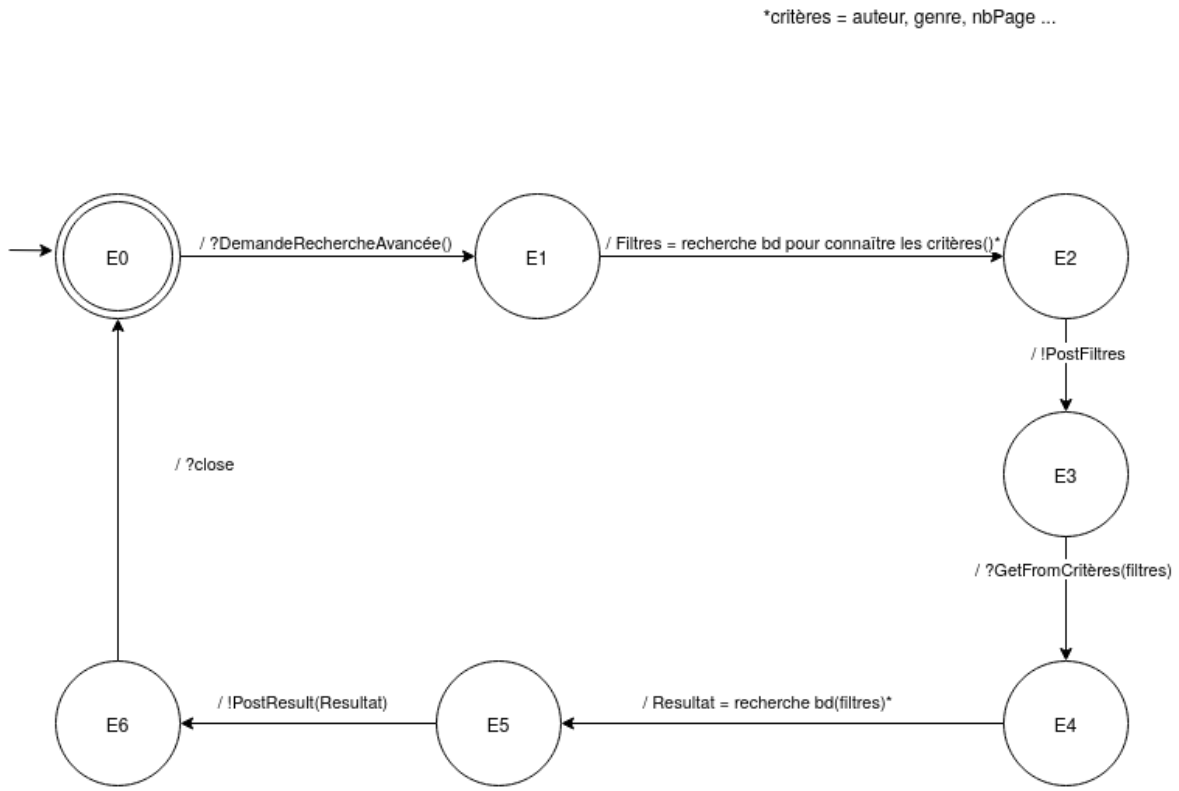

figure 4.

Fonctions figure 4 :
. recherche bd pour connaître les critères() : rend tous les critères disponible pour la recherche avancée
. recherche bd(filtres) : filtre les livres de la BD pour renvoyer ceux correspondant aux filtres

Spécification figure 4 :
Le client clique sur le bouton de recherche avancée, l'API lance une requête à la base de données pour récupérer tous les auteurs, genres, nombre de pages, ... Les données sont ensuite données au client, qui sélectionne les filtres voulus. L'API fait une requête à la base de données sur les champs choisis pour récupérer les livres correspondant. Le résultat est finalement envoyé au client.

C - Les microservices de sauvegarde de données

Les microservices d'enregistrement ont pour but de permettre à un client connecté de garder des livres dans des bibliothèques personnelles. Le client à donc accès à une page de connexion/inscription. Il peut enregistrer/supprimer un livre dans une bibliothèque personnelle. Il peut aussi créer, supprimer et modifier ses bibliothèques personnelles.

C.a - enregistrement/suppression d'un livre sur une bibliothèque personnelle

Voici la modélisation de l'enregistrement/suppression d'un livre dans une bibliothèque personnelle pour l'API :

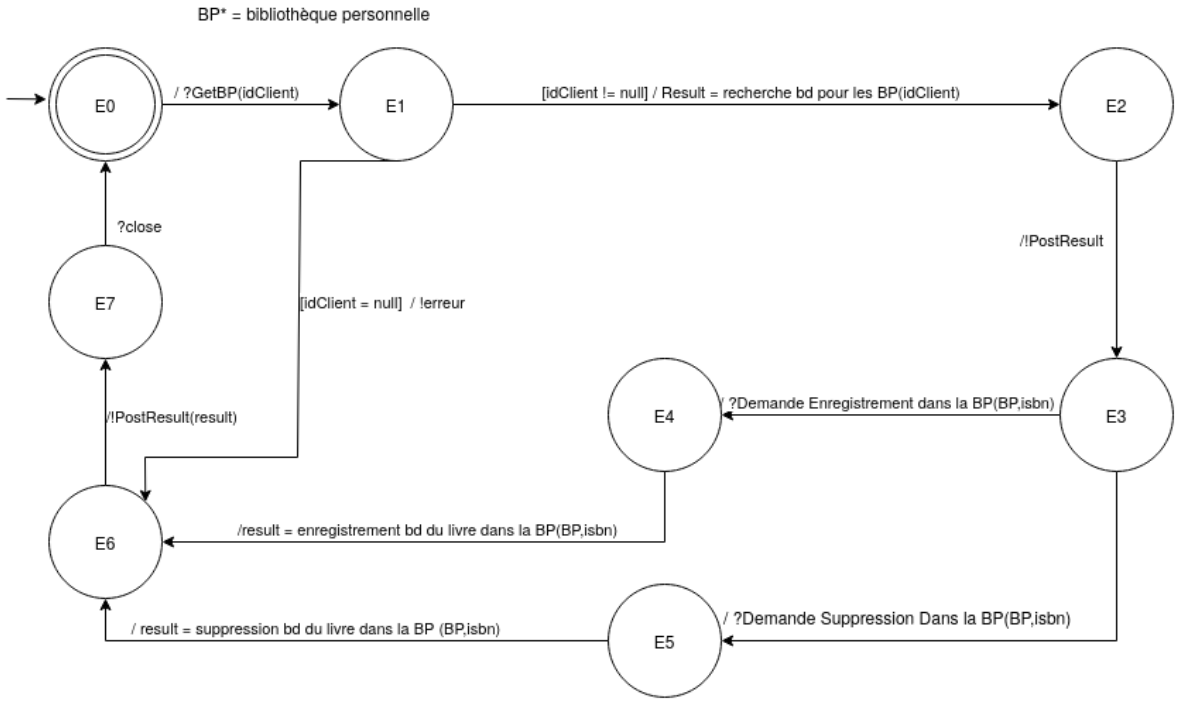

figure 5.

Fonctions figure 5 :

. recherche bd pour les BP(idClient) : récupère toutes les bibliothèques personnelles du client donné

. enregistrement bd du livre dans la BP(BP, isbn) : ajoute le livre isbn dans la bibliothèque personnelle BP si elle n'y est pas déjà

. suppression bd du livre dans la BP(BP,isbn) : retire le livre isbn de la bibliothèque personnelle BP s'il y est

Spécification figure 5 :
Quand le client clique sur le bouton ajout/suppression, il indique sur quelle bibliothèque personnelle il veut ajouter le livre ou de laquelle il va le retirer.
Pour cela, une requête est faite à l'API. L'API récupère les différentes bibliothèques personnelles du client depuis la base de données et les renvoie au client.
Pour ajouter un livre, le client choisit sa bibliothèque. L'information est récupérée par l'API qui la stocke dans la base de données. Si la bibliothèque n'existe pas ou si le livre est déjà enregistré dedans, un message d'erreur est renvoyé, dans le cas contraire un message de réussite du stockage est renvoyé.
Pour retirer le livre de la bibliothèque personnelle, l’API vérifie la présence du livre dans la bibliothèque puis fait une requête à la base de données pour le supprimer.

C.b - traitement des bibliothèques personnelles

Voici la modélisation du traitement des bibliothèques personnelles pour l'API :

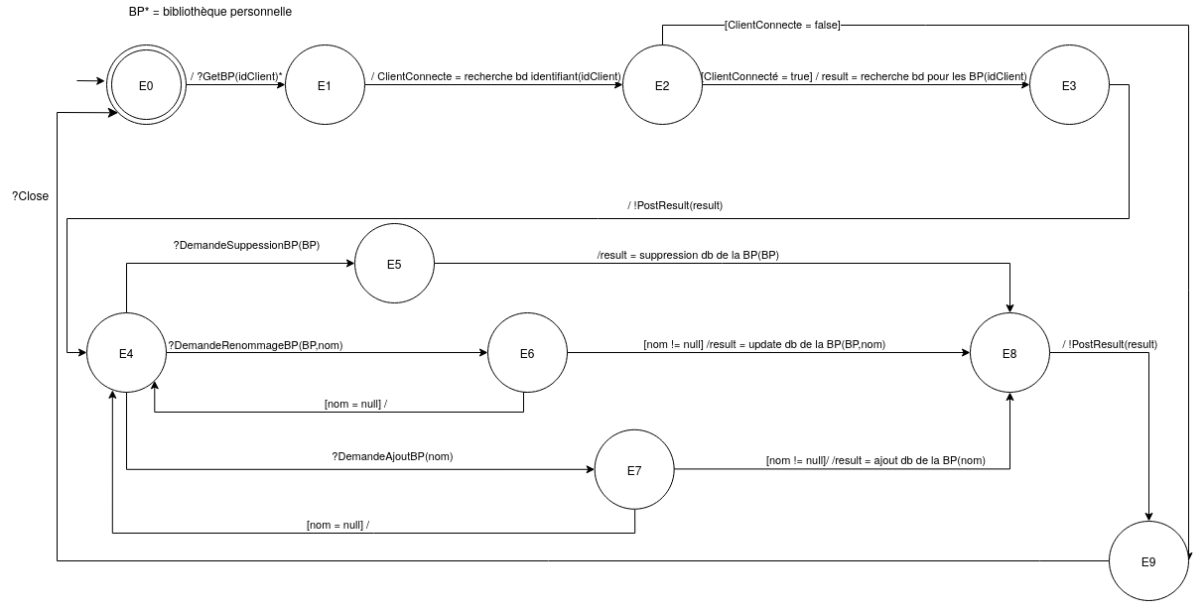

figure 6.

Fonctions figure 6 :
. recherche bd identifiant(idClient) : vérifie que l'idClient donné en paramètres existe dans notre base de clients
. recherche bd pour les BP(idClient) : présenté figure 5
. suppression db de la BP(BP) : supprime la bibliothèque personnelle BP de la base de données
. update db de la BP(BP,nom) : renomme la bibliothèque personnelle BP en nom
. ajout db de la BP(nom) : crée une bibliothèque personnelle nommée nom

Spécification figure 6 :
Quand le client est connecté, il a le choix entre ajouter, modifier ou supprimer des bibliothèques personnelles.
Tout d'abord, une requête est envoyée à l'API pour récupérer les différentes bibliothèques personnelles du client. L'API transmet la demande à la base de données, qui lui rend la liste des bibliothèques personnelles.
Le client a ensuite le choix entre supprimer, renommer ou ajouter une bibliothèque personnelle.

Supprimer : on vérifie que le client et la bibliothèque existent. On fait une requête avec la base de données pour la suppression. Le succès ou l'échec est renvoyé au client.

Renommer : on vérifie que le client et la bibliothèque existent et qu'un nouveau nom est donné. On fait une requête avec la base de données pour la modification. Les nouvelles propriétés de la bibliothèque sont renvoyées au client.

Ajouter : on vérifie que le client existe et qu'un nom est donné. On fait une requête avec la base de données pour l'ajout. Les informations de la nouvelle bibliothèque ou l’échec de l’opération est renvoyé au client.

3 - Automates et code

A - Les automates entre le code et l'analyse

Les automates ont permis de formaliser les spécifications et de faire une transition des idées de l'analyse vers les lignes de code. Les spécifications, écrites à la base pour l'analyse, ont été adaptées pour en faire des automates. Ces automates fournissent ensuite une structure pour le code.
Aucun des microservices présentés et représentés par les automates n'a vu sa logique interne changer pendant son implémentation dans le code. Certaines données comme le genre du livre n'étaient pas présentes dans les API en ligne et ont donc dû être remplacées pour l'implémentation.
Le plus gros changement par rapport aux spécifications est que les logiques liées à l'affichage du client ont dû être remplacées par les requêtes que fait le client suite à ses actions. Par exemple, on ne peut plus dire "quand le client clique sur un bouton" mais "quand le client fait x requête".

B - Les limites de l'automate

Bien que les automates offrent une modélisation de la logique métier derrière les différents microservices, il y a quand même eu un problème important pour leur implémentation : le MVC (la structure du projet).
Pour cause, les codes issus des automates sont généralement scindés dans différents fichiers, conformément à la logique MVC.
Les automates permettent donc d'avoir une représentation du code métier au global pour un microservice, mais sans documentation complémentaire il est difficile de comprendre l'implémentation. Cela peut poser problème à la création et à l'entretien du code.

4- Conclusion

En conclusion, les automates ont permis une transition de la phase d'analyse vers la phase de code. Ils constituent aussi une documentation claire et logique sur l'implémentation des codes métier. Ils restent toutefois insuffisants comme unique documentation pour comprendre le code final.

5- Références

Tous les documents ayant servis de références sont consultables dans le dossier "References" joint à ce rapport.

. *diapos_Langages-Automates*
- J. Christian Attiogbé

. *supplement_SAE_modelisation_automates_2025*
- J. Christian Attiogbé

. *exemple_rapport_1_2022-2023*
- N. Brevet, R. Gouraud, E. Garnier, M. Meunier

. *exemple_rapport_2_2022-2023*
- M. Bergeron, T. Freret, S. Baudouin, T. Martineau, F. Nicou